COMPUTING ONE-DIMENSIONAL GLOBAL MANIFOLDS OF POINCARÉ MAPS BY CONTINUATION

J.P. ENGLAND*, B. KRAUSKOPF*, AND H.M. OSINGA*

February 2005

Abstract. We present an algorithm to compute one-dimensional stable and unstable manifolds of saddle periodic orbits in a Poincaré section. The computation is set up as a boundary value problem by restricting the beginning and end points of orbit segments to the section. Starting from the periodic orbit itself, we use collocation routines from AUTO to continue the solutions of the boundary value problem such that one end point of the orbit segment varies along a part of the manifold that was already computed. In this way, the other end point of the orbit segment traces out a new piece of the manifold.

As opposed to standard methods that use shooting to compute the Poincaré map as the k-th return map, our approach defines the Poincaré map as the solution to a boundary value problem. This enables us to compute global manifolds through points where the flow is tangent to the section — a situation that is typically encountered unless one is dealing with a periodically forced system. Another major advantage of our approach is that it deals effectively with the problem of extreme sensitivity of the Poincaré map on its argument, which is a typical feature in the important class of slow-fast systems.

We illustrate and test our algorithm by computing stable and unstable manifold in three examples: the forced Van der Pol oscillator, a model of a semiconductor laser with optical injection, and a slow-fast chemical oscillator. All examples are accompanied by animations of how the manifolds grow during the computation.

Keywords: Poincaré map, stable and unstable manifolds, boundary value problem, slow-fast systems

AMS subject classification: 37D10, 37M20, 37C10, 65L10

1. Introduction. The study of global dynamics plays an increasingly important role in investigating the behavior of a dynamical system. Particularly in a parameter-dependent setting, a local analysis will provide information about bifurcations of steady states or periodic orbits, but global bifurcations typically involve global stable and unstable manifolds of these objects. Global invariant manifolds must be computed numerically and this remains a challenging and active area of research; see, for example, [20, 21] and references therein.

We consider here the problem of computing a one-dimensional stable or unstable manifold of a saddle point p_0 of a Poincaré map P of a vector field; see section 2 for formal definitions. The saddle point p_0 corresponds to a periodic orbit Γ that intersects the chosen Poincaré section Σ transversely. In a local neighborhood $U \subset \Sigma$ of p_0 the map P is a well-defined diffeomorphism, which is given as the (first) return of the trajectory through a given point $x \in U$ to the neighborhood U.

There are many algorithms available for computing one-dimensional stable and unstable manifolds of diffeomorphisms; see, for example, [12, 17, 24, 31] and references therein. While each algorithm has its own particular flavor, they all start from an approximation of the local stable or unstable manifold near the saddle point and then globalize or grow the manifold away from the saddle. In particular, the map must be well defined, which is clearly the case for explicitly defined diffeomorphisms, such as the Hénon map. These algorithms can also be used for Poincaré maps of vector fields, provided that two requirements are fulfilled:

*DEPARTMENT OF ENGINEERING MATHEMATICS, UNIVERSITY OF BRISTOL, BRISTOL BS8 1TR, UNITED KINGDOM

(i) The Poincaré map P is a well-defined diffeomorphism in a region of interest of phase space, and
(ii) $P(x)$ can be computed effectively and accurately by solving the initial value problem for the initial condition x.

Note that requirement (ii) means that $P(x)$ should not depend too sensitively on the argument x: if a change of x below the accuracy of the computation has an order-one effect on $P(x)$ then P cannot be computed accurately in the usual way by solving the initial value problem.

It is important to realize that requirement (i) is not globally satisfied in the case of a *general vector field without periodic forcing*: irrespective of the choice of the section Σ, the Poincaré map P is *not* a diffeomorphism on the whole of Σ. In fact, if a Poincaré map is a globally-defined diffeomorphism, then it is equivalent to the stroboscopic map of a periodically forced system [30]. Periodically forced vector fields are an important class of systems; examples include the forced Van der Pol and forced Duffing equations; see, e.g., [9, 27]. Indeed, the Poincaré map P of a periodically forced vector field is a diffeomorphism on the entire section Σ and can typically be computed accurately by integration over the forcing period. On the other hand, if the vector field under consideration is not periodically forced then P is only a local diffeomorphism near p_0, namely as long as the flow remains transverse to the section Σ. Further away from p_0 the flow must become tangent to Σ. At such a point the Poincaré map, defined by a fixed number of intersections with Σ, is not a diffeomorphism and typically not even continuous. The locus, which we denote by C, where the flow is tangent to Σ, divides Σ into different regions on which a local Poincaré map can be defined (as the return map to this part of the section). There may also be regions where P is undefined, meaning that points in such a region never return to Σ. Standard methods can be used to compute stable and unstable manifolds up until the discontinuity boundary C is reached, but will not be able to compute parts of the manifolds that cross C. It may be possible to continue such a computation by restarting the algorithm in the next region by redefining the Poincaré map (effectively by considering a different number of intersections with Σ). This is done, for example, in [3] where the algorithm of [17] was adapted, but the method is quite cumbersome and ad-hoc.

Requirement (ii) is not satisfied for the important class of *slow-fast systems*. In their simplest form, slow-fast systems have two different timescales, such that one variable changes much faster relative to the other. In fact, finding the saddle point p_0 (by computing the associated periodic orbit Γ) may already be very difficult. Furthermore, the evaluation of P by solving an initial value problem typically leads to numerical problems, even in a local neighborhood of the saddle p_0. As a result, the computation of global manifolds in slow-fast systems is extremely difficult and standard methods break down. It is not uncommon for the stretching/contraction of the manifold to be of a factor of 10^6 in these systems, such that a tiny pertubation in the initial condition may lead to a huge difference in the entire orbit segment — the orbit may not even return to the section.

In this paper we present an algorithm, called the *ManBVP algorithm*, that is able to compute the (un)stable manifold of a saddle point $p_0 \in \Sigma$ even if requirements (i) and/or (ii) are not satisfied. Specifically, it can deal with

- Poincaré maps that are not globally defined and feature discontinuity boundaries C in the region of interest, and

- Poincaré maps of slow-fast systems.

The key idea is to continue the entire orbit segment between x and $P(x)$ as a boundary value problem. The computation of an (un)stable manifold in Σ is started from the saddle periodic orbit Γ, which we find by continuation with AUTO [4]. The orbit segment is then continued using pseudo-arclength continuation and collocation routines from AUTO, not as a periodic orbit but as an orbit with both boundary points in Σ. The first boundary point of the orbit segment is initially varied along the (un)stable eigenvector of p_0 in Σ. In this way, the other boundary point of the orbit segment begins to trace out an initial piece of manifold. We then allow the first boundary point to vary along this initial piece and subsequently computed parts of the manifold, so that the other boundary point continues to trace out more and more of the (un)stable manifold. New points are added to the approximation of the manifold at varying distances according to the curvature of the manifold.

The ManBVP algorithm works even if requirement (i) is not satisfied because the boundary value problems we are solving, which also contain the integration time T of the orbit segment from x to $P(x)$, are continuous across C. Put bluntly, we are only interested in the fact that the end points of the orbit segments lie in Σ and do not care how many times the orbit segments intersect Σ in between. As the computation moves through C, the orbit used in the continuation automatically gains or loses an intersections with Σ. In the computation this corresponds to a change in direction of the first boundary point along the manifold already computed.

Our method also does not require (ii) to be satisfied because it continues the entire orbit segment from x to $P(x)$ (with the collocation routine of AUTO) from the given periodic orbit Γ. Since the distance between orbit segments in the continuation is measured along the entire orbit segment, the algorithm can deal effectively with even quite severe sensitivity on the initial condition, as one encounters in slow-fast systems; see section 6 for an example.

The ManBVP algorithm is not limited to systems that do not satisfy requirements (i) and/or (ii). It can be used in general to compute one-dimensional global manifold of Poincaré maps — just like any of the standard methods. However, our method is slightly more computationally expensive. The difference in computation time is actually less than one may think. After all, any method spends most of the time integrating the vector field to compute P. The ManBVP algorithm simply does this in a slightly more expensive way by solving/continuing a boundary value problem with the collocation solver of AUTO. An additional advantage is that the computed orbit segments give an impression of the two-dimensional global manifolds of the periodic orbit in the full phase space. This gives further insight into the geometry of the problem.

The performance of the ManBVP algorithm is illustrated in detail with three examples. Each example is accompanied by animations that show which orbit segments in the full phase space are used as the computation proceeds. First, we use the algorithm to compute stable and unstable manifolds in the periodically forced Van der Pol oscillator. This is a test case example where the result of the computation can be compared directly with standard methods. Secondly, we investigate an optically injected semiconductor laser, which gives rise to a non-globally defined Poincaré map. This is the example from [3] and we use it here to illustrate how our algorithm automatically computes global manifolds across the discontinuity boundary C. We also compute disconnected parts of the stable manifold that were not found before. Third, we compute stable and unstable manifolds of a slow-fast model of a chemical

oscillator as introduced in [1, 14]. To our knowledge, this is the first time that such a computation was performed in a slow-fast system that is characterized by extreme difference in the absolute sizes of the eigenvalues of the saddle. As a result, the orbit segments used during the computation are almost identical along about 85% of their length and only start to differ at the very end.

The outline of this paper is as follows. In section 2 we introduce some notation and give some background information. In section 3 we explain the ManBVP algorithm in detail for the case of an n-dimensional vector field. Section 4 shows its performance for the forced Van der Pol oscillator. In section 5 we consider the non-globally defined Poincaré map of an optically injected semiconductor laser. The example of the slow-fast chemical oscillator is discussed in section 6. Finally, we draw some conclusions and point to future work in section 7.

2. Notation and background. We consider a vector field

$$\dot{x} = f(x), \tag{2.1}$$

where $x \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}^n$ is sufficiently smooth. Of course, f may depend on parameters, but we are interested here in a single vector field where all parameters are fixed. Furthermore, suppose that (2.1) has a saddle periodic orbit Γ. That is, Γ is a periodic orbit with Floquet multipliers both inside and outside the unit circle of the complex plane, and only the trivial Floquet multiplier on the unit circle.

The Stable Manifold Theorem [23] guarantees the existence of global stable and unstable (immersed) manifolds that are as smooth as f. The global stable manifolds $W^s(\Gamma)$ and $W^u(\Gamma)$ are defined as the set of trajectories that tend to Γ in forward and backward time, respectively:

$$W^s(\Gamma) = \left\{x \in \mathbb{R}^n \mid \varphi^t(x) \to \Gamma \text{ as } t \to \infty\right\}$$

and

$$W^u(\Gamma) = \left\{x \in \mathbb{R}^n \mid \varphi^t(x) \to \Gamma \text{ as } t \to -\infty\right\}$$

In order to construct a Poincaré or first-return map one considers an $(n-1)$-dimensional Poincaré section Σ transverse to Γ. (The section Σ could be any smooth manifold transverse to Γ, but one usually works with $(n-1)$-dimensional hyperplanes.) Let p_0 be an intersection point of Γ with Σ. Due to transversality, there is a neighborhood $U \subset \Sigma$ of p_0 where we can define the Poincaré map in the usual way as

$$\begin{aligned} P : U &\to U \\ x &\mapsto \varphi^{t_x}(x) \end{aligned} \tag{2.2}$$

where φ^t is the flow of (2.1) and t_x is the time it takes to return to U for the first time. Considering the Poincaré map is a standard approach in dynamical systems theory, which has the advantage that the dimension of the system is reduced by one. The periodic orbit Γ in $\mathbb{R}^n$ corresponds to the point $p_0 \in U$, which is a hyperbolic saddle point of P. In U the Poincaré map is a diffeomorphism, that is, a differentiable map with a differentiable inverse. Note that $\varphi^{t_x}(x)$ will typically intersect Σ more than once, say k times, where all but the last intersection lie outside the neighborhood U. Therefore, it is a standard approach to define the Poincaré map P as the k-th intersection with Σ. This should not be confused with the fact that one speaks of P

as the first return map, which is meant with respect to return to the neighborhood U.

As was mentioned in the introduction, unless one considers a periodically forced system, the Poincaré map P is not a diffeomorphism on the entire section Σ. We define the discontinuity locus C as

$$C := \{x \in \Sigma \mid f(x).\mathbf{n} = 0\} \tag{2.3}$$

where $\mathbf{n}$ is the normal to Σ and the dot denotes the inner product in $\mathbb{R}^n$. (If Σ is a smooth manifold, rather than a hyperplane, then $\mathbf{n}$ depends on x.) The complement of C consists of open regions of Σ. In some of these regions it is possible to define a local Poincaré map as the k-th return to Σ for some positive integer k. However, there may be regions in which points never return to the section Σ under the flow φ^t, so that not even a local Poincaré map can be defined.

Since P is a local diffeomorphism near p_0, the Stable Manifold Theorem guarantees the existence of smooth (un)stable manifolds $W^s(p_0)$ and $W^u(p_0)$ in the region where P is a diffeomorphism. In this paper we are interested in what happens when $W^s(p_0)$ and $W^u(p_0)$ cross the discontinuity locus C. Therefore, we define (generalizations of) these manifolds as:

$$W^s(p_0) = W^s(\Gamma) \cap \Sigma$$

and

$$W^u(p_0) = W^u(\Gamma) \cap \Sigma.$$

Note that this definition, has the advantage of being very geometrical. It follows from the properties of the manifolds $W^s(\Gamma)$ and $W^u(\Gamma)$ in $\mathbb{R}^n$ that the (generalized) global manifold $W^s(p_0)$ and $W^u(p_0)$ exist and are a union of manifolds in Σ. The question of what $W^s(p_0)$ and $W^u(p_0)$ look like becomes one of singularity theory: what are the possible intersections of a smooth l-dimensional manifold in $\mathbb{R}^n$ with an $(n-1)$-dimensional hyperplane?

Even in this general setting, it is still easier to compute $W^s(p_0)$ and $W^u(p_0)$, rather than $W^s(\Gamma)$ and $W^u(\Gamma)$. To explain how this can be done we restrict to the case that $n = 3$ from now on, that is, we consider a two-dimensional Poincaré plane Σ in a three-dimensional phase space. This is an important special case, as is also shown by the examples in sections 4–6. Furthermore, it is much easier to imagine and visualize the underlying geometry. Note that now both $W^s(p_0)$ and $W^u(p_0)$ are (sets of) one-dimensional manifolds in Σ.

A simple example helps to explain the concepts. Suppose that there is a single saddle-periodic orbit $\Gamma \in \mathbb{R}^3$ that is convex (as is the case, for example, after a Hopf bifurcation). Any section Σ that intersects Γ transversely intersects the periodic orbit in two points, say, p_0 and p_1. Near both p_0 and p_1 one can define the Poincaré map P as a local diffeomorphism. In the simplest case, the set C is a single smooth curve where the flow is tangent to the section Σ. This curve divides the two regions of definition of P.

In this situation, the stable and unstable manifolds of p_0 and p_1 in Σ may cross C. One possibility is that a branch of $W^s(p_0)$ coincides with a branch of $W^s(p_1)$, effectively connecting p_0 and p_1; see already Figure 5.4. This corresponds in the three-dimensional phase space to the case that Σ intersects one 'side' of $W^s(\Gamma)$ in a single curve. There are other, more complicated possibilities of how Σ may intersect $W^s(\Gamma)$.

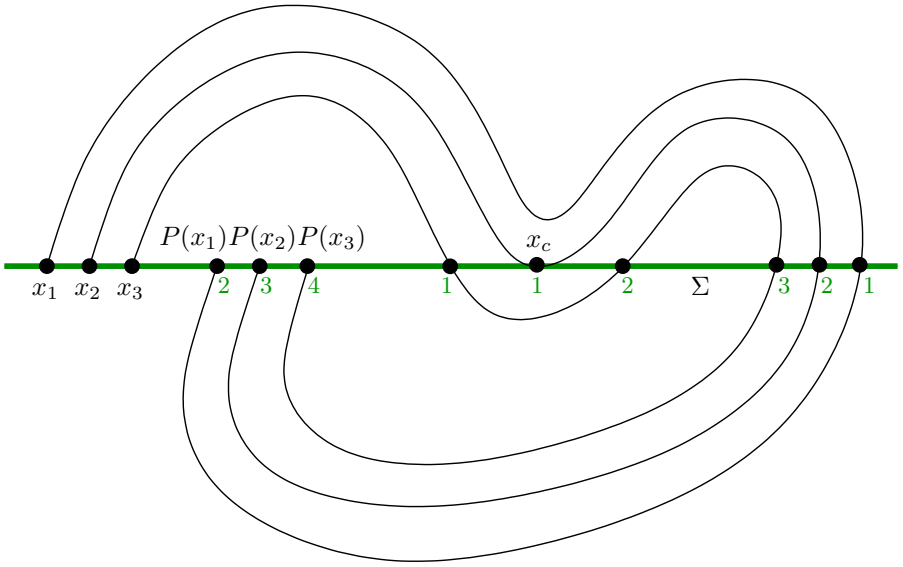


FIG. 2.1. *When defined as the second return to the section Σ, the Poincaré map is discontinuous at the tangency point $x_c \in C$. However, when P is continued as the boundary value problem (2.4) then it is continuous at x_c.*

For example, the intersection in Σ may consist of a closed curve that corresponds to a piece of $W^s(p_0)$ that is not connected to any saddle. While a systematic discussion of all possibilities is beyond the scope of this paper, section 5 gives a good impression of some of the possibilities.

A final, important point is how one can compute the Poincaré map P. Except in very special cases, there is no explicit form for P, so that it must be computed numerically. The most common technique to approximate P is by solving the initial value problem for x, that is, to integrate (2.1) until (the approximation of) $\varphi^{t_x}(x)$ lies again in $U \subset \Sigma$ for some t_x. Then $P(x) = \varphi^{t^*}(x)$ and $t_x = t^*$.

The point we take in this paper is that one should think of the Poincaré map not only as assigning $P(x)$, but as assigning the entire orbit segment $\{\varphi^t(x) \mid 0 \leq 0 \leq t_x\}$. This segment is the solution $\mathbf{u}$ of the two-point boundary value problem that solves (2.1) subject to the boundary conditions

$$\mathbf{u}(0) = x \in U \subset \Sigma \quad \text{and} \quad \mathbf{u}(T) \in U \subset \Sigma, \tag{2.4}$$

where $T = t_x$. The usual way of solving the initial value problem for x to find $P(x)$ can be interpreted as solving (2.4) by single shooting.

When solving the boundary value problem (2.4) by collocation with AUTO the integration time T is automatically solved for; see section 3 for details. The key advantage is that a curve of C with adjacent regions where the Poincaré map can be defined locally as the k-th return and the $(k+2)$-nd return to Σ, respectively, is no longer a discontinuity boundary in the space of boundary value problems. The situation is sketched in Figure 2.1. Therefore, the ManBVP algorithm is perfectly capable of computing global manifolds across such discontinuity curves.

Furthermore, the boundary value problem approach also deals efficiently with the problem of sensitivity of the integration with respect to the initial condition, which is a hall-mark of slow-fast systems. It is generally accepted that the best way of finding even a periodic orbit in a slow-fast system is to solve the associated boundary value

problem. An efficient and reliable method is Gauss collocation as used, for example, in AUTO [4] or CONTENT [22]; see also [10] for an alternative approach. We essentially take this insight to its logical conclusion and also use this boundary value problem approach for computing global manifolds. The ManBVP algorithm uses this same optimal theoretical setup. It is able to compute global manifolds in systems with very different time scale, of up to 8 orders of magnitude difference, as is demonstrated with the example in section 6.

3. Algorithm. We explain the ManBVP algorithm for the computation of a one-dimensional *unstable* manifold $W^u(p_0)$ of a saddle periodic orbit $p_0 \in \Sigma$; that is, p_0 has a single unstable eigenvalue $|\lambda^u| > 1$. The stable manifold can be computed in exactly the same way by reversing time. We assume that λ^u is greater than zero so that the Poincaré map is orientation preserving on $W^u(p_0)$; if $\lambda^u < 0$, then the periodic orbit Γ used to start the computation must be covered twice such that its period is doubled.

We calculate an approximation of $W^u(p_0)$ as an ordered list of mesh points $M = \{\phi_0, \phi_1, .., \phi_N\}$ up to a prescribed arclength A. In between mesh points we use linear line segments. The points ϕ_k lie in Σ at varying distance from each other depending on the curvature of the manifold. Our methods grows the manifold by computing one mesh point after another, and it can be seen as an adaptation of the method in [17] to the specific context of general Poincaré maps.

To compute points ϕ_k we follow entire orbit segments that define the Poincaré map by continuation of a two-point boundary value problem by starting from the periodic orbit Γ. The initial point of the orbit segment varies continuously, initially in the direction of the unstable eigenvector, along a part of the manifold that was already computed. The end point then traces out a new part of $W^u(p_0)$ in Σ. A new point ϕ_{k+1} is added to M whenever requirements on the distance between points are reached that depend on the curvature of $W^u(p_0)$. In this way, the algorithm automatically selects a minimum set of points for rendering the manifold in Σ according to prescribed accuracy criteria. A pseudo-code representation of how a branch of unstable manifold is grown is given in Figures 3.2 and 3.3. The main step of the ManBVP algorithm is illustrated in Figure 3.1.

To solve the boundary value problems we use the orthogonal collocation routines of AUTO [4]. We ensure that the computation is accurate by increasing the number of mesh points used in the collocation if necessary.

During a computation we can also record the orbit segments $\mathbf{u}_k$ (from Σ back to Σ) that are used to find the points $\phi_k = \mathbf{u}_k(1)$ that are added to M. These orbit segments give a good impression of the relevant part of the two-dimensional manifold $W^u(\Gamma)$ of the associated periodic orbit Γ of the vector field.

3.1. Computing the periodic orbit. We need to supply the algorithm with initial data before the start of a computation. Namely, we need a saddle periodic orbit that intersects the Poincaré section Σ at a point p_0, and we need the unstable eigendirection on Σ. An approximation for the periodic orbit $\Gamma = \{x(t) \mid 0 \le t \le T\}$, where $T > 0$ is the minimal period of the orbit, can be found either by direct analysis or, if the right-hand side f of (2.1) depends on a parameter, by continuation for example, from a Hopf bifurcation.

We use AUTO for the continuation of the two-point boundary value problem in our algorithm, which means that the total integration time must be scaled to the

interval [0,1]. Hence, the periodic orbit Γ is a solution of the system

$$\mathbf{u}'(t) = Tf(\mathbf{u}(t)), \tag{3.1}$$

with boundary condition

$$\mathbf{u}(1) - \mathbf{u}(0) = \mathbf{0}, \tag{3.2}$$

that is, $\Gamma = \{\mathbf{u}(t) \mid 0 \leq t \leq 1\} = \{x(tT) \mid 0 \leq t \leq 1\}$. The period T is now a parameter of the system. Note that an approximation of Γ obtained with AUTO will already be in this scaled form. Otherwise AUTO routines can be used to scale the integration time so that the problem formulation is in the required format.

To set up the computation in the Poincaré section Σ we need to shift the periodic orbit such that $\mathbf{u}(0) = \mathbf{u}(1) = p_0$, where p_0 is the intersection of Γ with Σ that we are concerned with. This is done by shifting all the mesh points of the discretized orbit such the last mesh point of the orbit is closest to Σ. Linear interpolation is then used to find a point on Σ, which is added to the mesh as the start and endpoint of the orbit. We correct this orbit using Newton's method in AUTO subject to the imposed boundary conditions,

$$\mathbf{u}(0) \in \Sigma, \tag{3.3}$$

$$\mathbf{u}(1) \in \Sigma. \tag{3.4}$$

This shifted and corrected orbit is denoted by $\mathbf{u}_0(t)$ and is one part of the starting data. Note that Γ typically intersects Σ in points other than p_0, but only p_0 is recognized as the point of interest, since $p_0 = \mathbf{u}_0(0) = \mathbf{u}_0(1)$.

3.2. Computing the eigenvector. We also use AUTO to find $E^u(p_0)$, the linear approximation of $W^u(p_0)$. Specifically, we find $E^u(p_0)$ as the intersection of the linear eigenspace $E^u(\Gamma)$ with Σ. To this end, we extend the system to compute the first variational equation

$$\begin{array}{rcll} \mathbf{u}'(t) & = & T\ f(\mathbf{u}(t)), & \mathbf{u} \in \mathbb{R}^n, \\ \mathbf{v}'(t) & = & T\ Df(\mathbf{u}(t))\ \mathbf{v}(t), & \mathbf{v} \in \mathbb{R}^n, \end{array} \tag{3.5}$$

where $\mathbf{u}'(t)$ is the same as (3.1) and $T\ Df(\mathbf{u}(t))$ is its Jacobian matrix. Here, we use the boundary condition (3.2) with the additional condition

$$\mathbf{v}(1) - \lambda_1 \mathbf{v}(0) = \mathbf{0}, \tag{3.6}$$

such that the orbit is periodic and the vector $\mathbf{v}$, which gives the direction of the derivative at each mesh point, extends in length by a factor λ_1 over one period of the orbit. Here, λ_1 is a free parameter, and a branching bifurcation takes place along the trivial solution branch $\mathbf{v} = \mathbf{0}$ at $\lambda_1 = \lambda^u$. We also require the so-called integral Floquet eigenfunction normalization by imposing that

$$\int_0^1 \mathbf{v}(t) * \mathbf{v}(t) - \lambda_2 dt = 0.$$

Here, λ_2 represents a 'distance' along the eigenfunction and is again a free parameter. Finally, if $\mathbf{u}(t)$ is a solution, $\mathbf{u}(t+\sigma)$ is also a solution for any σ. To ensure uniqueness of the periodic orbit, we use the phase condition

$$\int_0^1 \mathbf{u}(t) * \mathbf{u}'_{\text{old}}(t) dt = 0$$

where $\mathbf{u}_{\rm old}(t)$ is the previous solution computed in the continuation, which fixes σ.

The extended system (3.5) is initialized using the periodic orbit $\mathbf{u}_0(t)$ and the trivial vector bundle $\mathbf{0}$ for $\mathbf{v}$. The free parameters of the system are the period T, λ_1, which is initialized at an approximation of the unstable Floquet multiplier, and λ_2, which is initialized at zero. In the first run a branching bifurcation is detected at $\lambda_1 = \lambda^u$. We switch branches such that we continue along the eigenfunction $\mathbf{v}$ and continue the solution up to a pre-specified distance. This gives the unstable Floquet vectors at every mesh point of the periodic orbit forming a "ribbon" of $E^u(\Gamma)$ that intersects Σ.

The linear line field that defines $E^u(\Gamma)$ typically intersects Σ in a curve, rather than a straight line. The eigenspace $E^u(p_0)$ is obtained by projecting the direction of $E^u(\Gamma)$ with base point p_0 onto Σ. Note that $E^u(p_0)$ is tangent to $E^u(\Gamma) \cap \Sigma$ at p_0. In this paper we compute $E^u(p_0)$ by projection, which provides good starting data for all examples that we considered. However, it is also possible to compute a small segment of $E^u(\Gamma) \cap \Sigma$ as the initial approximation to $W^u(p_0)$; see [8] for a discussion in the context of delay differential equations.

3.3. Boundary conditions and user-defined functions. To set up notation, let

$$L_i(\tau) = (1-\tau)\,\phi_{i-1} + \tau\phi_i,\ 0 \le \tau \le 1.$$

denote the (parameterized) line segment between the already computed points ϕ_{i-1} and ϕ_i. In the course of the ManBVP algorithm we allow one boundary point to vary along $L_i(\tau)$, which means that the current solution $\mathbf{u}$ satisfies the additional boundary condition

$$\mathbf{u}(0) - L_i(\tau) = \mathbf{0}. \tag{3.7}$$

Furthermore, we require that $\mathbf{u}(1)$ lies in Σ, that is, we require (3.4). Note that (3.7) replaces (3.3), because it automatically ensures that $\mathbf{u}(0) \in \Sigma$.

As $\mathbf{u}(t)$ is continued, $\mathbf{u}(0)$ moves along the line segment $L_i(\tau)$ and $\mathbf{u}(1)$ traces out a new piece of manifold. In order to detect during a computation that the end of the line segment $L_i(\tau)$ has been reached we introduce the user-defined functions

$$\mathrm{UZ}(0) = \tau, \tag{3.8}$$

$$\mathrm{UZ}(1) = \tau - 1. \tag{3.9}$$

To decide when to generate the next point ϕ_{k+1} we consider the distance between the end boundary point and the last computed point, which we define as

$$\Delta_k = \|\mathbf{u}(1) - \phi_k\|. \tag{3.10}$$

Furthermore, we monitor the angle

$$\alpha_k = \angle(\phi_{k-1}, \phi_k, \mathbf{u}(1)) \tag{3.11}$$

between the last two computed points and the end boundary point of the current orbit. We pre-specify the accuracy conditions as in [17] and continue the orbit segment until $\alpha_k = \alpha_{\max}$ or $\Delta_k\alpha_k = (\Delta\alpha)_{\max}$. These conditions are evaluated as AUTO user-defined functions

$$\mathrm{UZ}(2) = \alpha_{\max} - \alpha_k, \tag{3.12}$$

$$\mathrm{UZ}(3) = (\Delta\alpha)_{\max} - \Delta_k\alpha_k. \tag{3.13}$$

For all the examples in this paper we use the accuracy conditions $\alpha_{\max} = 0.3$ and $(\Delta\alpha)_{\max} = 10^{-5}$.

As is discussed in [17], there may be sharp folds in the manifold that require an extremely small Δ_k, which is beyond computational numerical accuracy limitations. Therefore, we specify the user-defined function

$$\text{UZ}(4) = \Delta_{\min} - \Delta_k. \tag{3.14}$$

If we detect UZ(4)=0 as $\mathbf{u}(1)$ traces out the manifold, and either $\alpha_k > \alpha_{\max}$ or $\Delta_k\alpha_k > (\Delta\alpha)_{\max}$, then we accept the endpoint of the orbit segment as a new mesh point in M and give a warning message to the user. For the examples in this paper we used $\Delta_{\min} = 10^{-4}$.

The accuracy control of the ManBVP algorithm is performed by monitoring the same user-defined accuracy conditions as the method for diffeomorphisms in [17]. The additional computational error from the boundary value problem can be controlled independently. Therefore, the overall accuracy of a manifold computation up to a prescribed arclength can be estimated in the same way. Note that this estimate depends on — generally unknown — Lipschitz constants, so that no a priori bounds are available. As is common in the field, the accuracy of a computation can be checked by comparison with a computation of increased accuracy.

3.4. Starting the computation. The computation starts from the periodic orbit Γ in the form $\mathbf{u}_0$, which satisfies the boundary conditions (3.4) and (3.7). Note that we no longer require boundary condition (3.2).

We begin the continuation with $\mathbf{u}(t) = \mathbf{u}_0(t)$, set $\phi_0 = p_0$, and allow $\mathbf{u}(0)$ to vary along $E^u(p_0)$. We temporarily define, for the initial continuation only, the point ϕ_1 as $\phi_0 + \delta_{\max} v_0$, where v_0 is the eigendirection spanning $E^u(p_0)$ and $\delta_{\max}$ is small and specified by the user. With this definition of ϕ_1 the initial line segment $L_1(\tau)$ is well defined. In this setup, τ varies in the interval [0,1] such that the start boundary point of the orbit segment moves from p_0 along the unstable eigendirection up to a small distance $\delta_{\max}$. The free continuation parameters are τ, the distance along $E^u(p_0)$, and T, the period of the orbit. Initially, $\tau = 0$, $\mathbf{u}(0) = \mathbf{u}(1) = p_0$ and T is the period of $\mathbf{u}_0$, that is, of the period of Γ.

We also need to reformulate the accuracy conditions slightly differently for the first continuation away from p_0. We define the distance

$$\Delta_1 = \|\mathbf{u}(1) - \mathbf{u}(0)\|$$

and the angle

$$\alpha_1 = \angle(p_0, \mathbf{u}(0), \mathbf{u}(1)),$$

That is, we use the begin and the end boundary points of the solution $\mathbf{u}$. When either of the accuracy conditions are met then the continuation stops and we set $\phi_1 = \mathbf{u}(0)$ and $\phi_2 = \mathbf{u}(1)$. We now have the first three points of our approximation of $W^u(p_0)$, which are added to the list M. We also have the orbit segment whose end point is the last computed point ϕ_2 stored in memory as $\mathbf{u}_2(t) = \mathbf{u}(t)$.

To find the next point ϕ_3, we simply vary $\mathbf{u}(0)$ along the line segment $L_2(\tau)$ defined in the usual way by ϕ_1 and ϕ_2. Since the current solution $\mathbf{u}(t)$ has its start point at ϕ_1 and its end point at ϕ_2, we set $\tau = 0$.

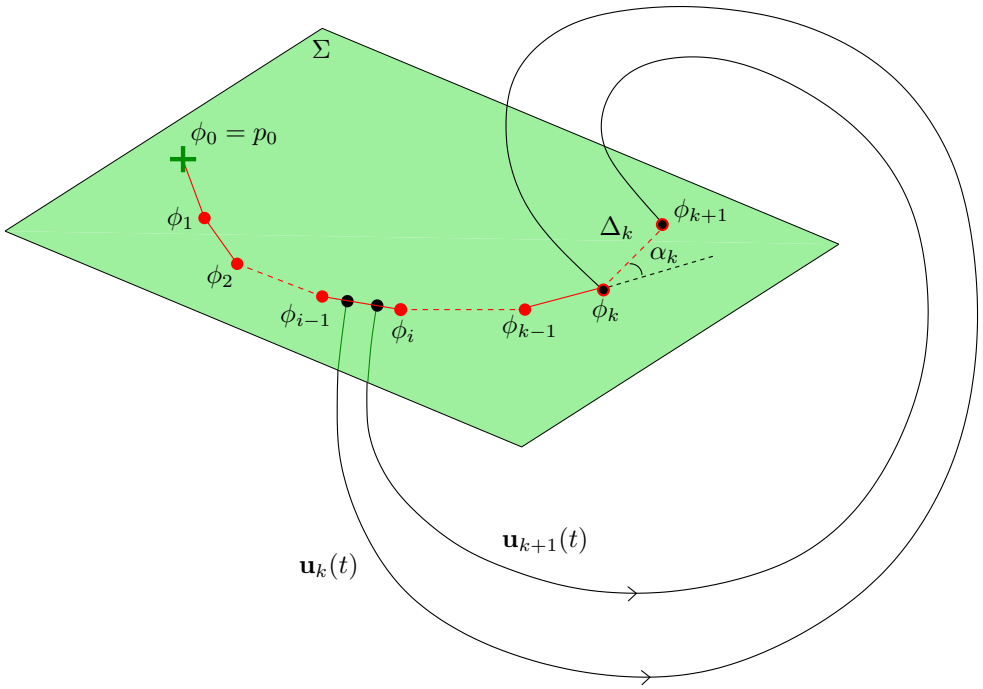


FIG. 3.1. *Graphical illustration of the ManBVP algorithm. A new point ϕ_{k+1} is found by continuation of the known orbit segment $\mathbf{u}_k(t)$ with $\mathbf{u}_k(0) \in L_i(\tau)$ and $\mathbf{u}_k(1) = \phi_k$ until $\mathbf{u}_{k+1}$ is found with the property that its endpoint lies at distance Δ_k from ϕ_k and either $\alpha_k = \alpha_{\max}$ or $\Delta_k \alpha_k = (\Delta\alpha)_{\max}$.*

3.5. The general step of finding ϕ_{k+1}. A general step of the algorithm consists of continuing solutions of the boundary value problem (3.4) and (3.7) by calling the respective AUTO routines from the main program. Recall that this means that the end point $\mathbf{u}(0)$ lies in the line segment $L_i(\tau)$, that is, between ϕ_{i-1} and ϕ_i. The computation stops when one of the user defined functions UZ(0)–UZ(4) that are monitored has a zero; see (3.8), (3.9), (3.12), (3.13), (3.14).

If either UZ(0) or UZ(1) is zero, then the end point $\mathbf{u}(0)$ tracing the computed piece of $W^u(p_0)$ has reached the end of the interval $L_i(\tau)$. In this case the computation is restarted by allowing this end point to be in $L_{i-1}(\tau)$ or $L_{i+1}(\tau)$, respectively. It is possible that during the computation the end of L_k, that is, of the computed mesh, is reached. In this case we add $\phi_{k+1} = \mathbf{u}(1)$ as the new mesh point to M, which ends this step. This situation typically arises at the very beginning of a computation, because the initial point ϕ_2 (as found in section 3.4) is still very close to the saddle point p_0. The above step is often repeated, which effectively means that the point ϕ_2 is iterated under the Poincaré map until the distance between iterates is large enough. On the other hand, it may be that $W^u(p_0)$ converges to an attracting fixed point, which is detected when the boundary point $\mathbf{u}(0)$ reaches the end of the computed mesh *and* $\mathbf{u}(1)$ is at distance less then $\Delta_{\min}$ from $\phi_k = \mathbf{u}(0)$. In this case the algorithm stops. Similarly, the end of L_0 may be reached, which means that the boundary point $\mathbf{u}(0)$ has returned to the fixed point p_0. In this case the computation stops; for an example see section 5.

If either UZ(2) or UZ(3) is zero then the other end point $\mathbf{u}(1)$ lies at the desired distance from the last point ϕ_k in M, according to the prescribed accuracy conditions (3.12)–(3.13). We define $\phi_{k+1} = \mathbf{u}(1)$ and add this point to the list M. This finishes the step. The next point ϕ_{k+2} is then found in the same way, by starting the continuation from the present solution $\mathbf{u}$, using L_i with the present τ-value in $[0, 1]$ (that is not equal to either 0 or 1).

As was mentioned in section 3.3, a zero of UZ(4) may be detected when the manifold folds so sharply that the distance between ϕ_k and ϕ_{k+1} would be below the minimal prescribed distance $\Delta_{\min}$. If this is detected, we accept $\phi_{k+1} = \mathbf{u}(1)$ as a new point of M, while giving a warning message to the user.

It is typical that the solution $\mathbf{u}$ becomes much longer and more complicated during the computation of $W^u(p_0)$. To avoid problems of convergence of the collocation routine of AUTO we automatically adapt the number of mesh points along the solution when AUTO fails to converge to a solution after the prescribed number of Newton iterations.

The algorithm stops when the total required arclength is reached, or if the manifold converges to an attracting fixed point of the Poincaré map (which is detected as was explained above).

Global Variables:

Γ	the saddle periodic orbit $\{\mathbf{u}_0(t) \mid 0 \le t \le 1\}$.
p_0	intersection point of Γ with the Poincaré section Σ. (saddle of Poincaré map.)
v_0	unstable eigenvector of p_0.

ManBVP (Maximum distance between p_0 and ϕ_1: $\delta_{\max}$, target arclength: A**)**
  $\mathbf{u}(t) = \mathbf{u}_0(t)$;
  $(\phi_{i-1}, \phi_i) = (p_0, p_0 + \delta_{\max} v_0$;
  $\tau = 0$;
  $(\mathbf{u}_2(t), \tau)$ = BVP_Step($\mathbf{u}(t)$, ϕ_{i-1}, ϕ_i, p_0, **null**, τ);
  $\phi_0 = p_0$;
  $\phi_1 = \mathbf{u}_2(0)$;
  $\phi_2 = \mathbf{u}_2(1)$;
  $M = \{\phi_0, \phi_1, \phi_2\}$;
  $arclength = \|\phi_2 - \phi_1\| + \|\phi_1 - \phi_0\|$;
  $\mathbf{u}(t) = \mathbf{u}_2(t)$;
  $\tau = 0$;
  while $(arclength < A)$
    ϕ_k = last point in M;
    ϕ_{k-1} = next to last point in M;
    repeat
      $(\mathbf{u}(t), \tau)$ = BVP_Step($\mathbf{u}(t)$,ϕ_{i-1},ϕ_i, ϕ_{k-1},ϕ_k, τ);
      if $(\tau = 1)$ **then**
        $(\phi_{i-1}, \phi_i) = (\phi_i, \phi_{i+1})$; /* switch to line segment $[\phi_i, \phi_{i+1}]$ */
        $\tau = 0$;
      else if $(\tau = 0)$ **then**
        $(\phi_{i-1}, \phi_i) = (\phi_{i-2}, \phi_{i-1})$; /* switch to line segment $[\phi_{i-2}, \phi_{i-1}]$ */
        $\tau = 1$;
      end if
    until $(\tau \neq 0$ and $\tau \neq 1)$
    $\mathbf{u}_{k+1}(t) = \mathbf{u}(t)$;
    $\phi_{k+1} = \mathbf{u}_{k+1}(1)$;
    append(M, ϕ_{k+1});
    $arclength = arclength + \|\phi_{k+1} - \phi_k\|$;
  end while
  return M;
end.

FIG. 3.2. *A pseudo-code representation of the* ManBVP *algorithm, which calls* BVP_Step *in Figure 3.3 for the continuation of the orbit segment with* $\mathbf{u}(0)$ *varying over* (ϕ_{i-1}, ϕ_i).

BVP_Step (Current stored orbit: $\mathbf{u}(t)$, ϕ_{i-1}, ϕ_i, ϕ_{k-1}, ϕ_k, τ**)**
    AUTO Continuation Free Parameters:
        ICP(0) = τ;
        ICP(1) = Period T;
    AUTO Continuation boundary conditions:
        $L_i(\tau) = (1-\tau)\phi_{i-1} + \tau\phi_i$;
        $\mathbf{u}(0) \in L_i(\tau)$;
        $\mathbf{u}(0) \in \Sigma$;
        $\mathbf{u}(1) \in \Sigma$;
    AUTO Continuation user-defined functions:
        if (ϕ_k = **null**) **then**
            $\phi_k = \mathbf{u}(0)$; /* First run, use $\mathbf{u}(0)$ to compute α */
        end if
        $\alpha_k = \angle(\phi_{k-1}, \phi_k, \mathbf{u}(1))$; /* angle between ϕ_{k-1}, ϕ_k and $\mathbf{u}(1)$. */
        $\Delta_k = \|\mathbf{u}(1) - \phi_k\|$; /* distance between last two points */
        UZ(0) = τ;
        UZ(1) = $\tau - 1$;
        UZ(2) = $\alpha_{\max} - \alpha_k$;
        UZ(3) = $(\Delta\alpha)_{\max} - \Delta_k\alpha_k$;
        UZ(4) = $\Delta_{\min} - \Delta_k$;
    while (UZ(0) > 0 and UZ(1) < 0) /* $0 \le \tau \le 1$ */
        $\mathbf{u}$ = AUTO Continuation Step;
        Update free parameters and user-defined functions;
        if (UZ(2) = 0 or UZ(3) = 0)
            /* Reached limit of accuracy conditions */
            break;
        end if
        if (UZ(4) = 0 and (UZ(2) < 0 or UZ(3) < 0))
            /* Accuracy conditions already exceeded when $\Delta = \Delta_{\min}$,
                Accept point anyway. */
            print warning: "Accuracy conditions exceeded at $\Delta = \Delta_{\min}$";
            break;
        end if
    end while
    return ($\mathbf{u}(t)$, τ);
end.

FIG. 3.3. *The* BVP_STEP *routine, which uses* AUTO *continuation and collocation to find the orbit segment* $\mathbf{u}(t)$*, where* $\mathbf{u}(0)$ *varies along the line segment* $L_i(\tau)$*; see also Figure 3.2.*

4. The Forced Van der Pol oscillator. Our first example we consider as a test case of a globally-defined Poincaré map, namely that of the forced Van der Pol oscillator, a well-known periodically driven system; see [9, 11, 25]. The system can be written in the form of a three-dimensional vector field as

$$\begin{aligned} \dot{x} &= y, \\ \dot{y} &= -x - \alpha\left(x^2 - 1\right) y + \beta \cos \omega t, \\ \dot{t} &= 1, \end{aligned} \tag{4.1}$$

where α, β and ω are positive real parameters. The original unforced Van der Pol oscillator, with $\beta = 0$, describes an RLC-circuit with a nonlinear resistor that gives rise to oscillations in voltage and current; see [11] as a general reference. Here, α describes how quickly the slow phase recovers, β is the amplitude of the forcing and ω the forcing frequency.

We consider here the Poincaré map P, defined as the stroboscopic map taken every period $2\pi/\omega$ of the forcing term. As for any periodically forced system, this Poincaré map is globally defined on the sections $\Sigma_k = \{t = 2k\pi/\omega, k \in \mathbb{Z}\}$ (which can be identified). For $\alpha = 0.4$, $\beta = 0.375$ and $\omega = 0.893$, a saddle periodic orbit Γ exists that intersects Σ_k at $p_0 \approx (1.178, -0.772)$. We compute the one-dimensional stable and unstable manifolds $W^s(p_0)$ and $W^u(p_0)$ in the Poincaré section Σ_0 for the forced Van der Pol oscillator as a test case for the ManBVP algorithm. For this globally defined Poincaré map the manifolds could be computed with any one-dimensional manifold algorithm. The parameters are chosen such that the computed manifolds may be directly compared with those computed in [18].

Figure 4.1(a) shows $W^u(p_0)$ (red) and $W^s(p_0)$ (blue) computed with the ManBVP algorithm. A sink is located approximately at $(x, y) = (2.062, -0.098)$; this is the intersection of an attracting periodic orbit with Σ_k, and is indicated by a blue triangle. A repellor is the intersection of a repelling periodic orbit with Σ_k, indicated by the red square at approximately $(x, y) = (0.952, -0.785)$. Both branches of $W^u(p_0)$ converge to the sink and the left branch of $W^s(p_0)$ spirals in toward the repellor. Figure 4.1(b) shows all the orbit segments that were used in the computation of $W^s(p_0)$ and $W^u(p_0)$, together with the sections Σ_{-1}, Σ_0 and Σ_{+1} (green planes). During the computation of $W^u(p_0)$, the boundary point $\mathbf{u}(0)$ of each of the orbit segments lie in Σ_0 and the boundary points $\mathbf{u}(1)$ lie in Σ_{+1}. For the computation of $W^s(p_0)$ time is reversed, so that for the orbit segments used to compute $W^s(p_0)$, the boundary point $\mathbf{u}(1)$ lies in Σ_{-1}. As is common, the computed points on Σ_{+1} and Σ_{-1} are used on Σ_0 and added to $W^u(p_0)$ and $W^s(p_0)$, respectively.

Figure 4.2 further illustrates the unstable manifold computation by showing the orbit segments associated with the mesh points used to compute $W^u(p_0)$. The saddle periodic orbit Γ is the green curve that intersects both sections at p_0. The boundary point $\mathbf{u}(0)$ of an orbit segment is allowed to vary in Σ_0 along the section of $W^u(p_0)$ that was already computed. The boundary point $\mathbf{u}(1)$ of the orbit segment then traces out a new piece of manifold in Σ_{+1}. The orbit segments shown are those corresponding to the computed points on $W^u(p_0)$; they vary in color from dark red to light red in the computation. Notice how the orbit segments at the end of each branch tend to the attracting periodic orbit (cyan curve). The figure also gives a good impression of the two-dimensional unstable manifold $W^u(\Gamma)$, which forms a cusped torus for the chosen parameters; compare [18, Figure 5]. Figure 4.2(b) gives an enlarged view near Σ_{+1} and clearly shows how the points $\mathbf{u}(1)$ trace out $W^u(p_0)$. Notice also the repelling periodic orbit (red) inside the torus.

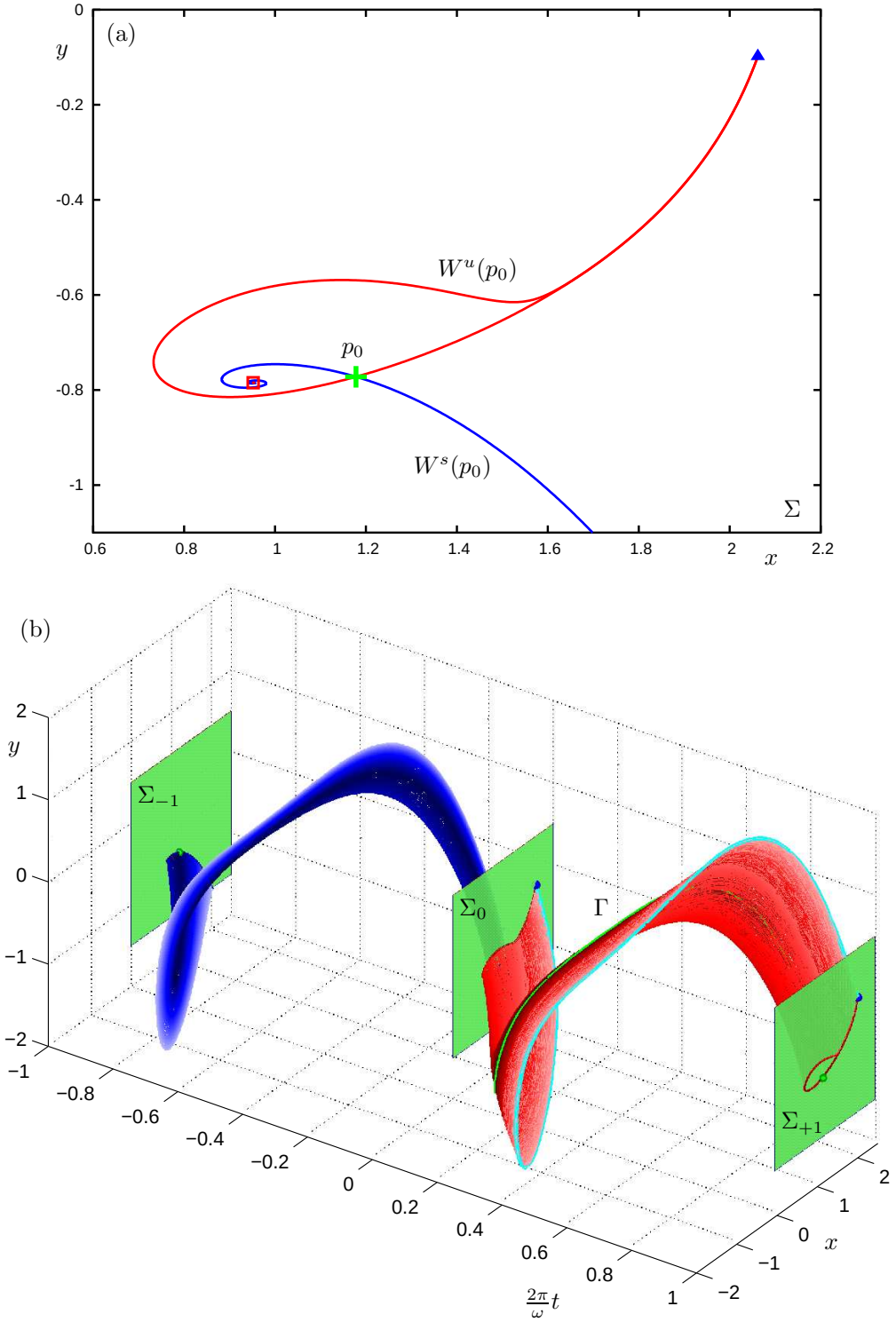


FIG. 4.1. *The global manifolds of system (4.1) for* $\alpha = 0.4$, $\beta = 0.375$ *and* $\omega = 0.893$. *Panel (a) shows the saddle* p_0 *and it unstable and stable manifolds,* $W^u(p_0)$ *(red) and* $W^s(p_0)$ *(blue), respectively. One side of* $W^s(p_0)$ *spirals into a source (red square) and both sides of* $W^u(p_0)$ *connect at a sink (blue triangle). The orbit segments that are associated with the mesh points generated by the ManBVP algorithm are shown in panel (b).*

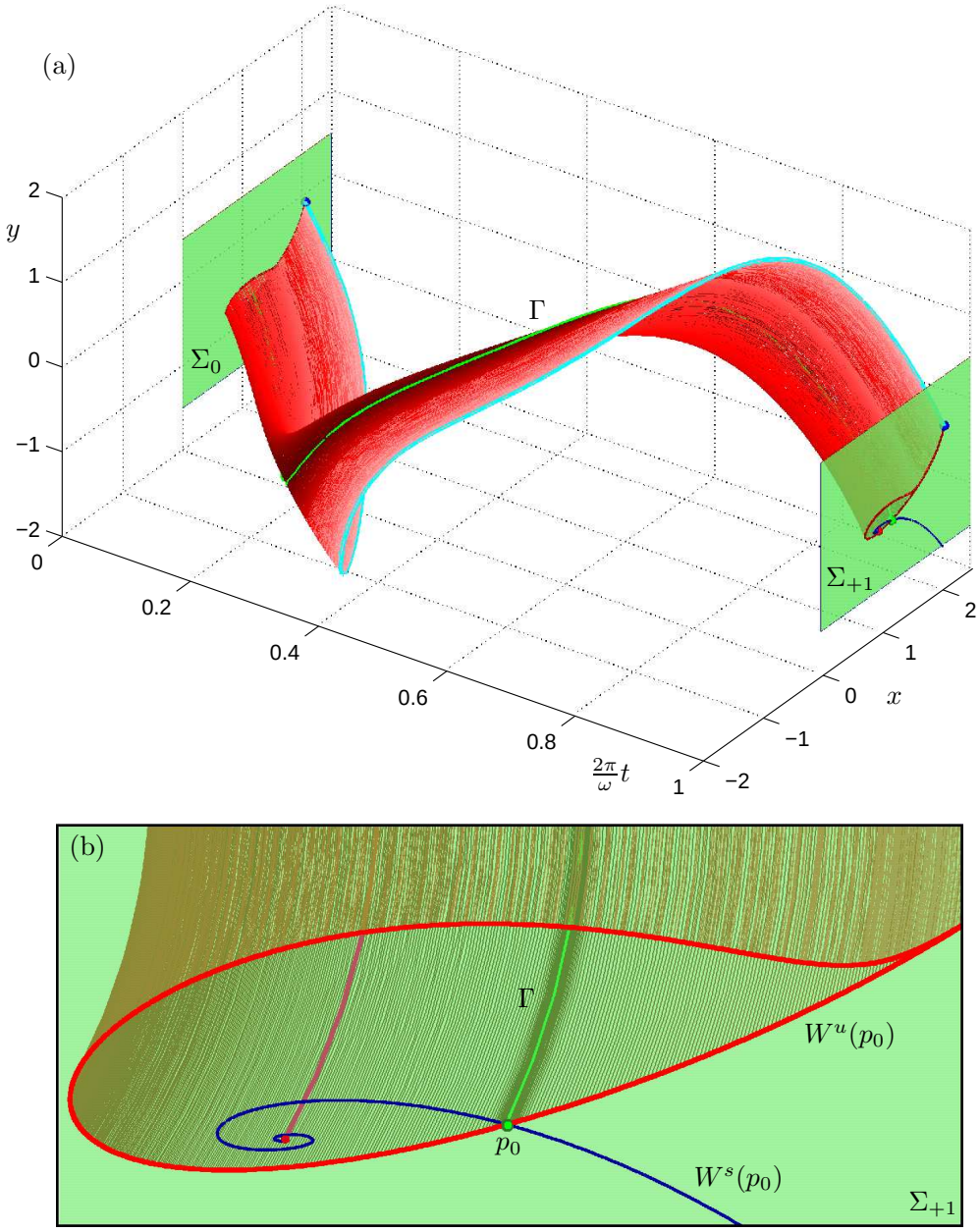


FIG. 4.2. *The orbit segments on $W^u(\Gamma)$ that are used in the computation of $W^u(p_0)$ are shown in panel (a). The boundary points $\mathbf{u}(0)$ and $\mathbf{u}(1)$ of the orbit segments on $W^u(\Gamma)$ lie in Σ_0 and Σ_{+1}, respectively. The color of the orbit segments on $W^u(\Gamma)$ varies from dark red to light red as the computation progresses. The attracting periodic orbit is shown in cyan and its boundary points in Σ_0 and Σ_{+1} are indicated by a blue point. An enlargement near the saddle p_0 (green) and the source (red) along with their orbit segments in Σ_{+1} is shown in panel (b). Compare also with Figure 4.1. See also the animation* eko_a01.gif.

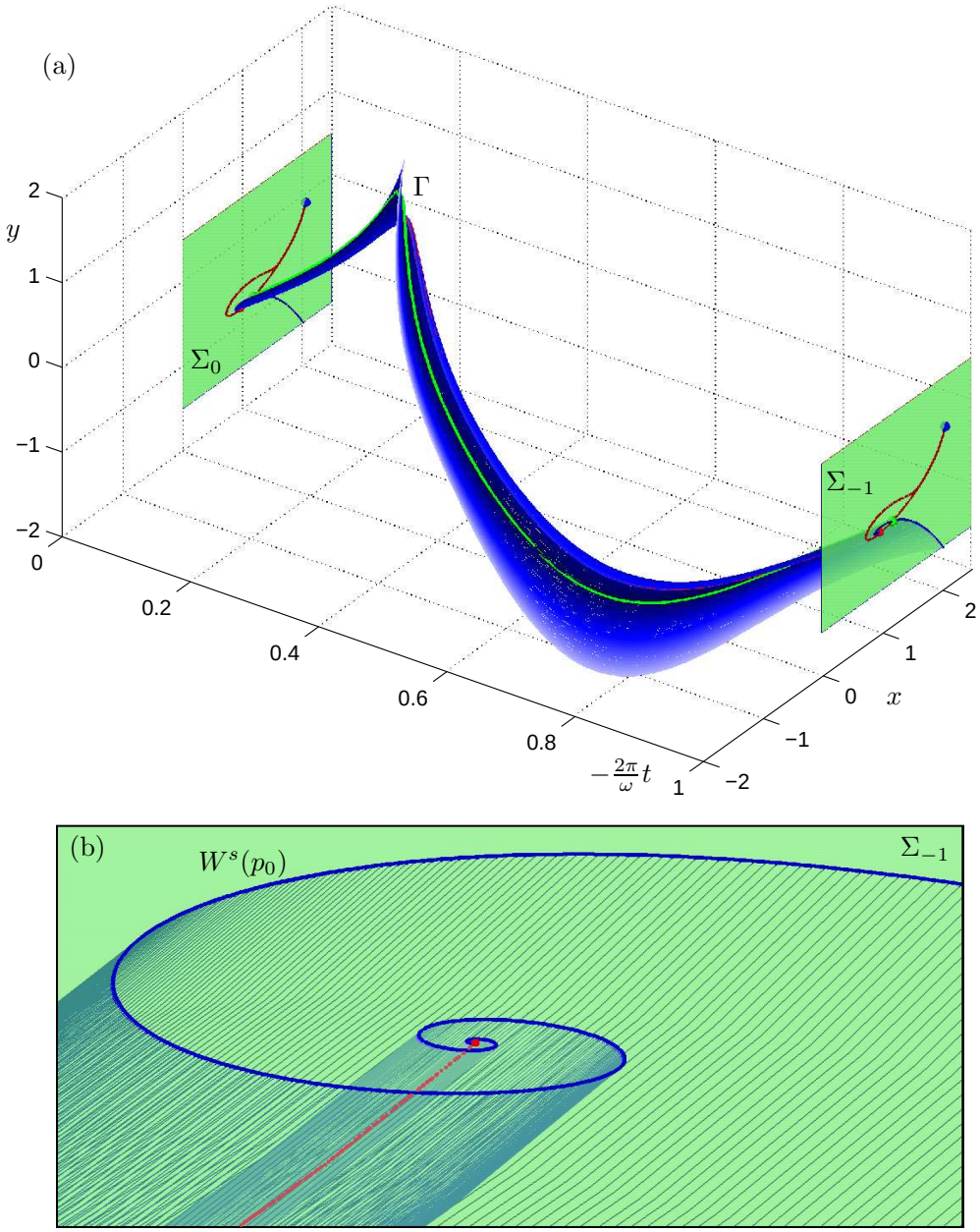


FIG. 4.3. *The orbit segments on $W^s(\Gamma)$ that are used in the computation of $W^s(p_0)$ are shown in panel (a). The boundary points $\mathbf{u}(0)$ and $\mathbf{u}(1)$ of the orbit segments on $W^s(\Gamma)$ lie in Σ_0 and Σ_{-1}, respectively. The color of the orbit segments on $W^s(\Gamma)$ vary from dark blue to light blue as the computation progresses. The unstable periodic orbit is shown in red and its boundary points in Σ_0 and Σ_{-1} are indicated by a red point. An enlargement near the saddle p_0 (green point) and the source (red point) in Σ_{-1} is shown in panel (b). Compare also with Figure 4.1. See also the animation* eko_a02.gif.

Figure 4.3 further illustrates the computation of $W^s(p_0)$. Orbits on the stable manifold accumulate on the saddle in forward time. Therefore, to grow the manifold from the saddle we need to reverse time. This means that orbit segments now flow from Σ_0 to Σ_{-1}. As $\mathbf{u}(0)$ of an orbit segment is allowed to vary along a section of $W^s(p_0)$ in Σ_0 that has already been computed, $\mathbf{u}(1)$ traces out a new piece of $W^s(p_0)$ in Σ_{-1}; the orbit segments vary in color from dark blue to light blue as the computation progresses. An enlarged view near Σ_{+1} is shown in Figure 4.3(b) to highlight how $W^s(p_0)$ spirals in toward the repelling periodic orbit (red).

5. Semiconductor laser with optical injection. In this section we investigate a semiconductor laser with optical injection as an example of a system that does not have a globally defined Poincaré map. Optical injection gives rise to rich dynamics in the laser including oscillations in the electric field and even chaos; see, for example, [28, 29]. The laser is modeled by the so-called rate equations

$$\begin{aligned} \dot{E} &= K + \left(\tfrac{1}{2}\left(1 + i\alpha\right) n - i\omega\right) E, \\ \dot{n} &= -2\gamma n - \left(1 + 2Bn\right)\left(|E|^2 - 1\right), \end{aligned} \tag{5.1}$$

where $E = (E_x, E_y)$ is the complex electric field and n is the population inversion (number of electron-hole pairs) [28]. The injected field strength is given by K and the detuning (between the frequency of the free-running laser and the injected frequency) by ω. The material properties of the laser are given by α, B and γ. We use the parameters $\alpha = 2$, $B = 0.015$, $\gamma = 0.035$, and choose $\omega = 0.270$ and $K = 0.290$, as in [3]. We study the Poincaré map P on the section given by the plane $\Sigma = \{(E, n) : n = 0\}$. A saddle periodic orbit Γ intersects the section four times at $\{p_0, p_1, p_2, p_3\}$, which we interpret as four-periodic points of P when P is the first return to Σ. The Poincaré map is discontinuous on the circle C where the flow is tangent to the section, given by $|E| = 1$. For $|E| < 1$ the flow through the section is upward (in the direction $n > 0$) and for $|E| > 1$ the flow is downward (in the direction $n < 0$).

The stable and unstable manifolds were computed in [3] with an adaptation of the algorithm of Krauskopf and Osinga [17]. This adaptation enabled the manifolds to be computed past C. This was done in [3] in quite an ad-hoc way by changing the number of intersections with Σ in the computation of the Poincaré map by shooting. Our method is able to compute a manifold automatically past points where P is discontinuous, that is, where the manifold crosses C. During the continuation the boundary conditions only require that the end points of the orbit segment lie in the section and it does not matter how many times the orbit segment intersects Σ in between. As the point $\mathbf{u}(1)$ of the solution of the boundary value problem passes through C, the solution changes continuously into an orbit segment with one fewer or one extra intersection and, effectively, the number of intersections with Σ changes automatically; see already Figure 5.3.

Figure 5.1(a) shows all manifolds that we computed for system (5.1) in the Poincaré section Σ with the ManBVP algorithm. The saddles $\{p_0, p_1, p_2, p_3\}$ are indicated by green crosses and a period-four sink is indicated by blue triangles. For clarity, some of the manifolds are highlighted in Fig. 5.1(b)–(d). Figure 5.1(b) shows the stable manifolds that spiral in toward single points of the Poincaré map P. The two branches starting from p_1 and p_3 never interact with the discontinuity curve C. Here, every point on the manifold is associated with an orbit segment that returns to Σ exactly four times. The two branches from p_0 and p_2, on the other hand, cross the curve C several times. A similar observation can be made in Figure 5.1(c) where branches of the stable manifold cross C. Note that for the branches shown here $W^s(p_0) = W^s(p_3)$

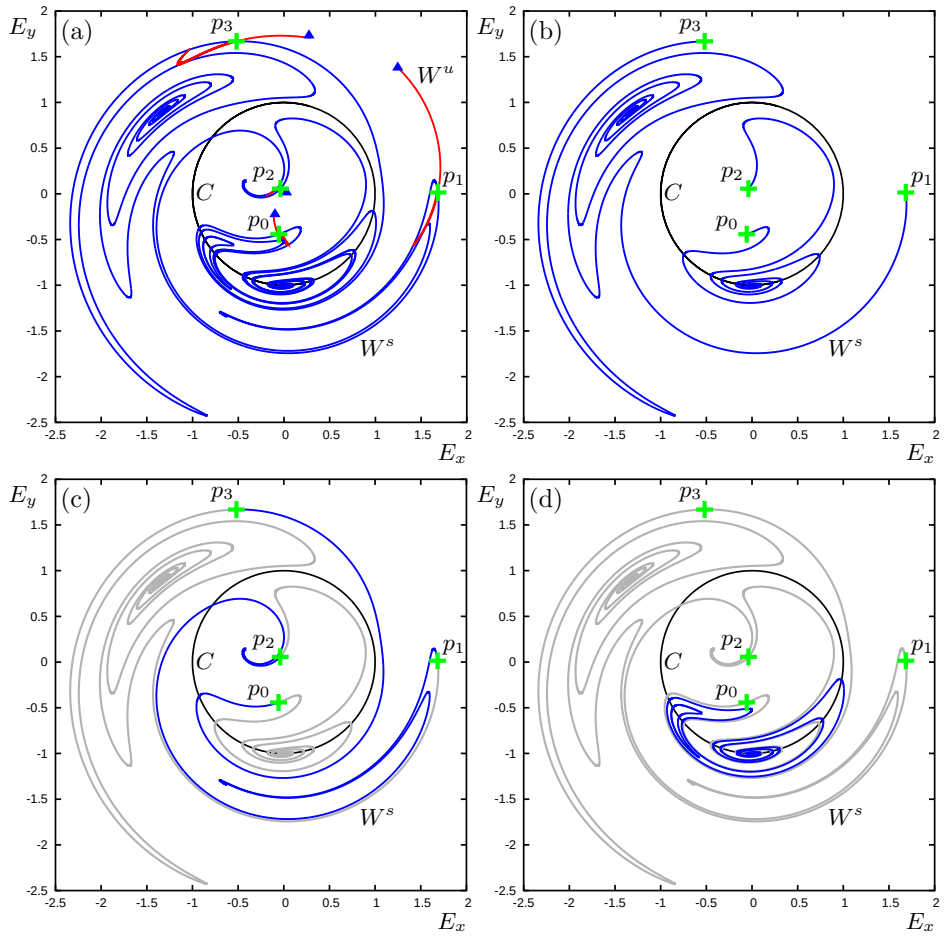


FIG. 5.1. *The stable and unstable manifolds, as computed by the ManBVP algorithm, of the four-periodic saddle* $\{p_0, p_1, p_2, p_3\}$ *(green crosses), of (5.1) in the Poincaré section* $\{n = 0\}$ *for parameters* $\alpha = 2$, $B = 0.015$, $\omega = 0.270$, $K = 0.290$ *and* $\Gamma = 0.035$. *The Poincaré map is discontinuous along the unit circle* C *(black curve). Panel (a) shows all stable and unstable manifolds connected to the periodic orbit and also two disjoint branches of the stable manifold. Panel (b) highlights just the stable manifolds that spiral into single points. Panel (c) highlights the branch of stable manifolds that join two of the saddle points. Panel (d) highlights two (of many) disjoint pieces of manifold.*

and $W^s(p_1) = W^s(p_2)$, that is, these branches connect different saddles. Figure 5.1(d) shows another property of the manifolds that can only happen for non-globally defined Poincaré maps. Namely, this panel shows two branches of the stable manifold that are not connected to any of the four saddle points. The computation of each of these different branches is discussed in more detail below.

5.1. Unstable manifolds. The unstable manifolds of all the saddle points are shown in Figure 5.1(a). A standard algorithm for the computation of unstable manifolds would define P as the fourth intersection with Σ so that each of these four-periodic points becomes a fixed point of the system. The ManBVP algorithm starts from the periodic orbit Γ in the full system and it is not necessary to identify how

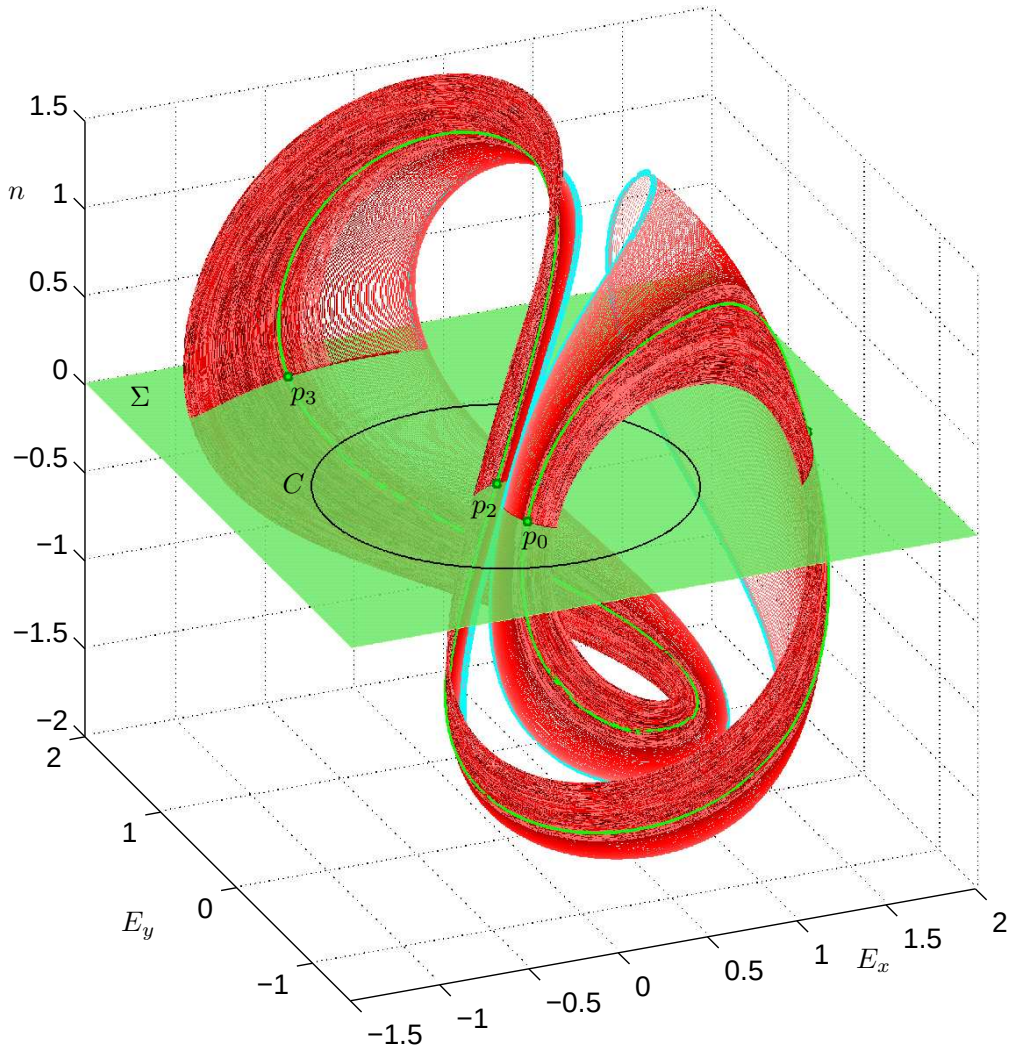


FIG. 5.2. *The unstable manifolds of the points* $\{p_0, p_1, p_2, p_3\}$ *as computed in* Σ. *The orbit segments that are used in the computation of* $W^u(p_3)$ *(red curves) give an impression of the entire two-dimensional manifold of* Γ *(green curve). One side of the unstable manifold tends to a period-two attractor (cyan curve), whilst the other side accumulates on a chaotic attractor.*

often Γ intersects Σ. Figure 5.2 shows the orbit segments used in computing $W^u(p_3)$; the computations of the other unstable manifolds are similar. The orbit color varies from dark red to light red as the computation of each branch progresses. The orbit segments give an impression of the entire two-dimensional unstable manifold $W^u(\Gamma)$. One side of $W^u(\Gamma)$ tends to a stable periodic orbit, whose intersections with Σ are the period-four sinks of the first-return map. The other side of $W^u(\Gamma)$ accumulates on a chaotic attractor, which is the reason for the 'randomness' in the color of the orbit segments. As illustrated in Figure 5.2 the unstable manifolds do not interact with the discontinuity boundary C so that they can be computed with a different method.

5.2. Crossing the discontinuity boundary. Let us now focus on the left branch of $W^s(p_0)$ as shown in Figure 5.1(c). This is more clearly illustrated in Figure 5.3. The saddle periodic orbit Γ is shown in green and all the orbit segments on $W^s(\Gamma)$ are blue. Row (a) shows all the orbit segments used in the computation of an initial piece of the stable manifold before it intersects C together with the time profile of the last orbit segment computed. Since, we are computing a stable manifold, we change the direction of the flow. Observe that all the orbit segments start near p_0 and (in negative time) flow downward from Σ. They next intersect Σ near p_3 and so on until they return to Σ for the fourth time. The boundary points $\mathbf{u}(1)$ of the orbit segments form the initial piece of $W^s(p_0)$. The time profile in Figure 5.3(a2) shows three intersections with Σ (green line) between $\mathbf{u}(0)$ and $\mathbf{u}(1)$; note that the end of the orbit segment is flowing downward into the section. Row (b) in Figure 5.3 shows the computation up to the point where the manifold intersects C. The time profile in panel (b2) shows that the last computed orbit segment still has three intersections with the section between $\mathbf{u}(0)$ and $\mathbf{u}(1)$, but now the orbit segment is tangent to Σ at the boundary point $\mathbf{u}(1)$. Row (c) in Figure 5.3 shows the orbit segments used in the computation after the manifold has been computed past C. The time profile in panel (c2) shows that there are now four intersections of the last computed orbit segment with Σ between $\mathbf{u}(0)$ and $\mathbf{u}(1)$.

Comparing Figure 5.3(a2) and (c2), the two orbit segments are very similar and the crossing of C should be viewed as a continuous transition by continuation in integration time: the orbit in Figure 5.3(c2) is similar to the orbit segment in Figure 5.3(a2), but integrated further. This means that, before reaching C, $\mathbf{u}(0)$ moves away from p_0 as $\mathbf{u}(1)$ traces out the end of the manifold, but after crossing C, $\mathbf{u}(0)$ moves back along $W^s(p_0)$ toward p_0. The method of [3] computed the stable manifolds past C by manually changing the number of intersections with Σ in the computation of P. Our method has the advantage of being able to compute past C automatically; a crossing of C is noted and marked by a change in the direction in which $\mathbf{u}(0)$ moves along W^s.

Figure 5.4 shows the orbit segments used in the entire computation of this branch of $W^s(p_0)$. As we already observe in Figure 5.1(c), this branch of $W^s(p_0)$ actually connects to p_3, such that $W^s(p_0) = W^s(p_3)$. The connection of two stable manifolds may at first glance appear to be counter-intuitive, but we must remember that p_0 and p_3 are part of the same periodic orbit Γ and that we are actually computing the intersection of $W^s(\Gamma)$ with Σ.

It its also possible for a manifold to have multiple intersection with C. Figure 5.1(b) shows four different branches of stable manifold that spiral in toward a single point. Figure 5.5 shows the orbit segments used in the computation of the right branch of $W^s(p_2)$. It turns out that the two-dimensional manifold $W^u(\Gamma)$ spirals around the stable manifold of an equilibrium; the point in Σ at the center of the spiral is the point where the one-dimensional stable manifold of this equilibrium intersects Σ. It is actually a point where the Poincaré map is undefined, as this point never returns to Σ. After crossing C the first time, the direction in which $\mathbf{u}(0)$ is moving along $W^s(p_2)$ changes and $\mathbf{u}(0)$ moves back toward p_2. When the manifold crosses C again, the direction of $\mathbf{u}(0)$ changes again. As the manifold spirals in, it crosses C many times, and each time, the orbit segment used in the continuation gains an extra intersection with Σ and the direction of $\mathbf{u}(0)$ changes along $W^s(p_2)$. This can be seen in the enlarged view in Figure 5.5(b). The orbit color varies from dark blue to light blue as the computation progresses. Observe that $\mathbf{u}(0)$ traverses back and forth along the

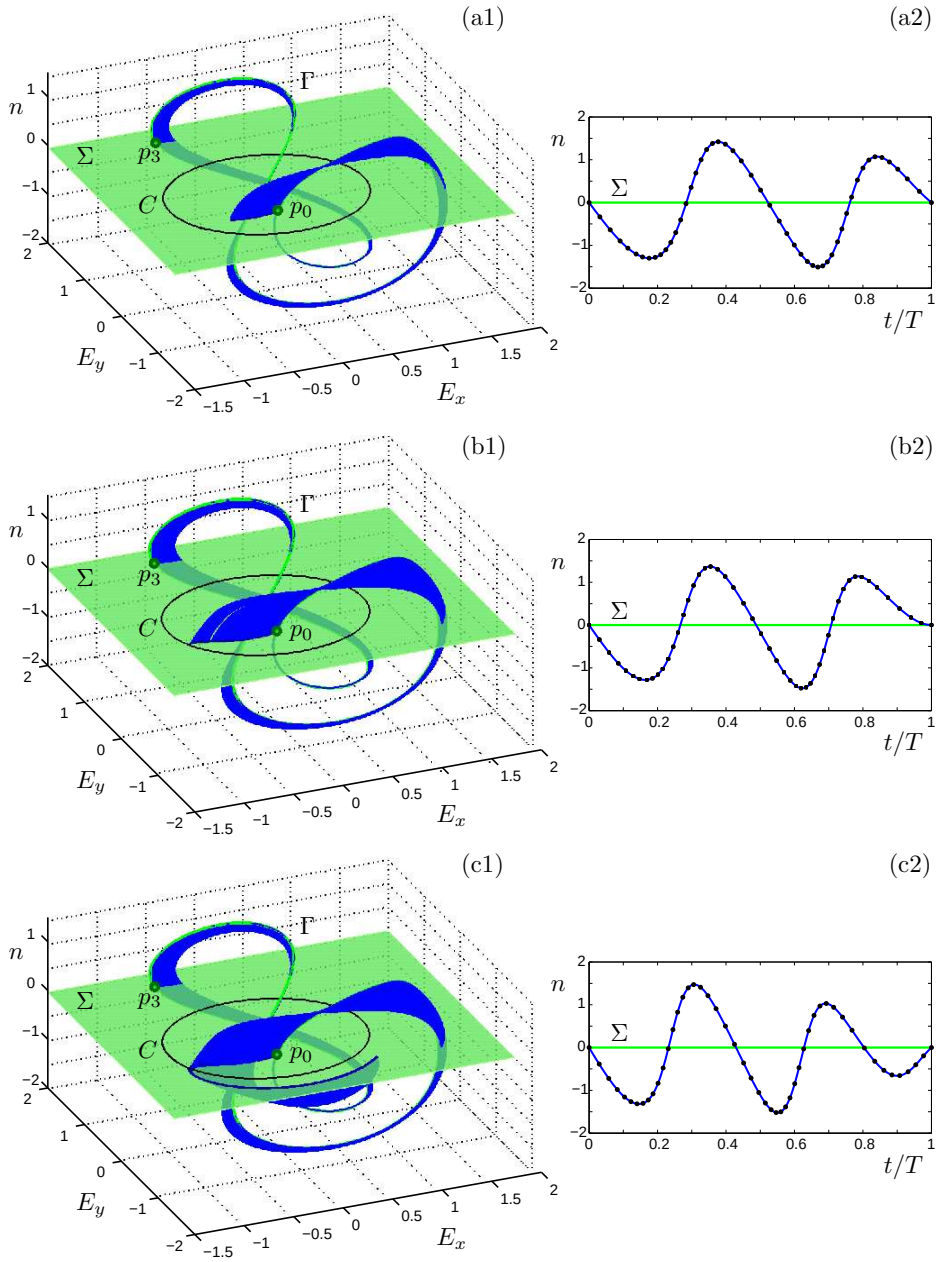


FIG. 5.3. *A demonstration of how $W^s(p_0)$ of system (5.1) is computed across C. Rows (a)–(c) show the computations before, at, and after the intersection with C, respectively. Panels in the left column show orbit segments on $W^s(\Gamma)$ used to compute $W^s(p_0)$. The time profiles of the last orbit segment computed in each case are shown in the right column (blue line), where the black dots indicate the mesh points and the Poincaré section Σ is the green line.*

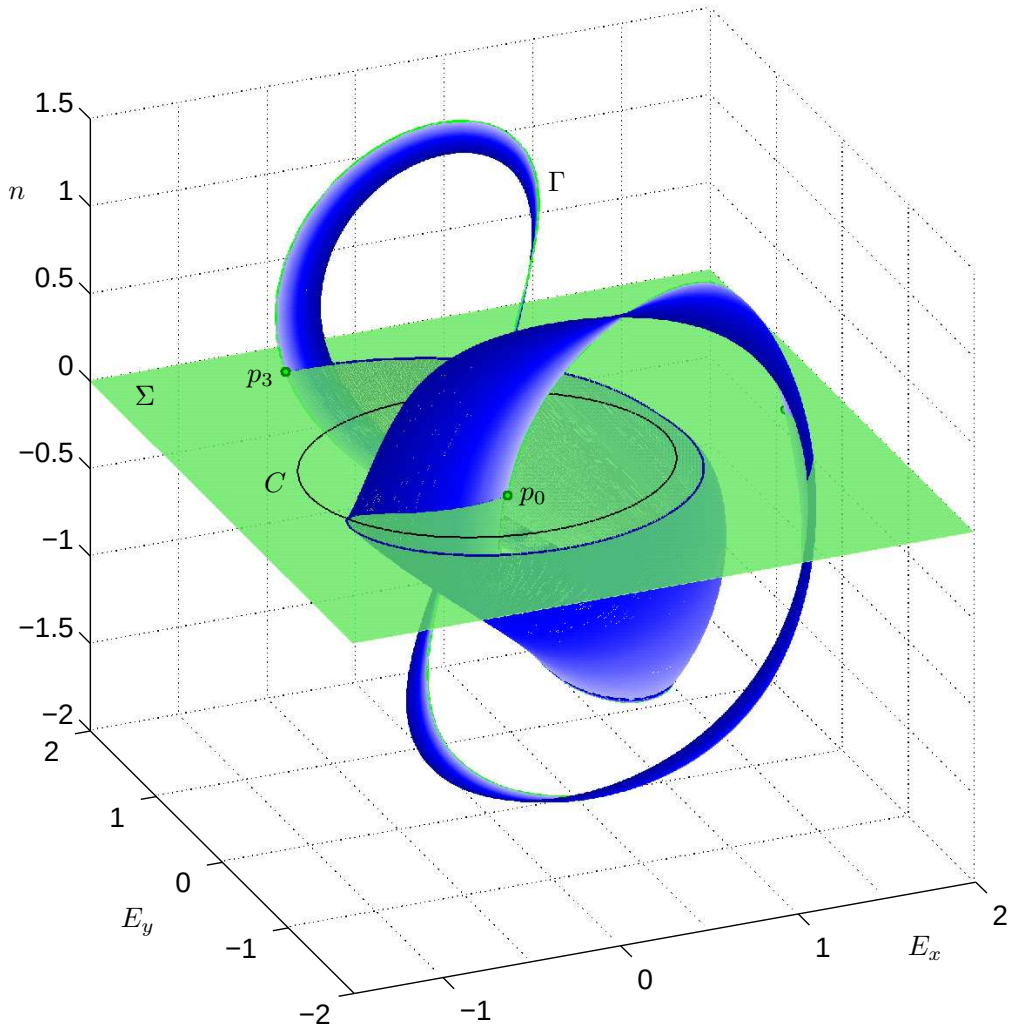


FIG. 5.4. *The entire branch of $W^s(p_0)$ that connects to p_3; compare Figure 5.3. Before $W^s(p_0)$ reaches C the orbit segment comes in from above Σ; after $W^s(p_0)$ has been grown past C, the orbit segment has an extra intersection and comes in from below Σ. See also the animation* eko_a03.gif.

manifold, which is indicated by the color of the orbit segments getting lighter as the manifold spirals inward. The initial points of all orbit segments, whose end boundary points make up most of the spiral, lie in a small segment of $W^s(p_2)$ and the same initial point is 'shared' by many of these orbit segments.

5.3. Disjoint branches. Figure 5.6(a) shows an enlarged view of one of the disjoint pieces of W^s that was shown in Figure 5.1(d). These disjoint pieces of manifold map under P to a piece of manifold that is connected to a saddle. In Figure 5.6(a) the entire disjoint piece of W^s maps to the blue piece of $W^s(p_1)$, such that its images traverses the manifold back and forth as W^s crosses C. This can be seen clearly in Figure 5.6(b), where all orbit segments have $\mathbf{u}(0)$ on $W^s(p_1)$ and $\mathbf{u}(1)$ on the disjoint manifold. This part of W^s is an isolated submanifold that is due to the way $W^s(\Gamma)$

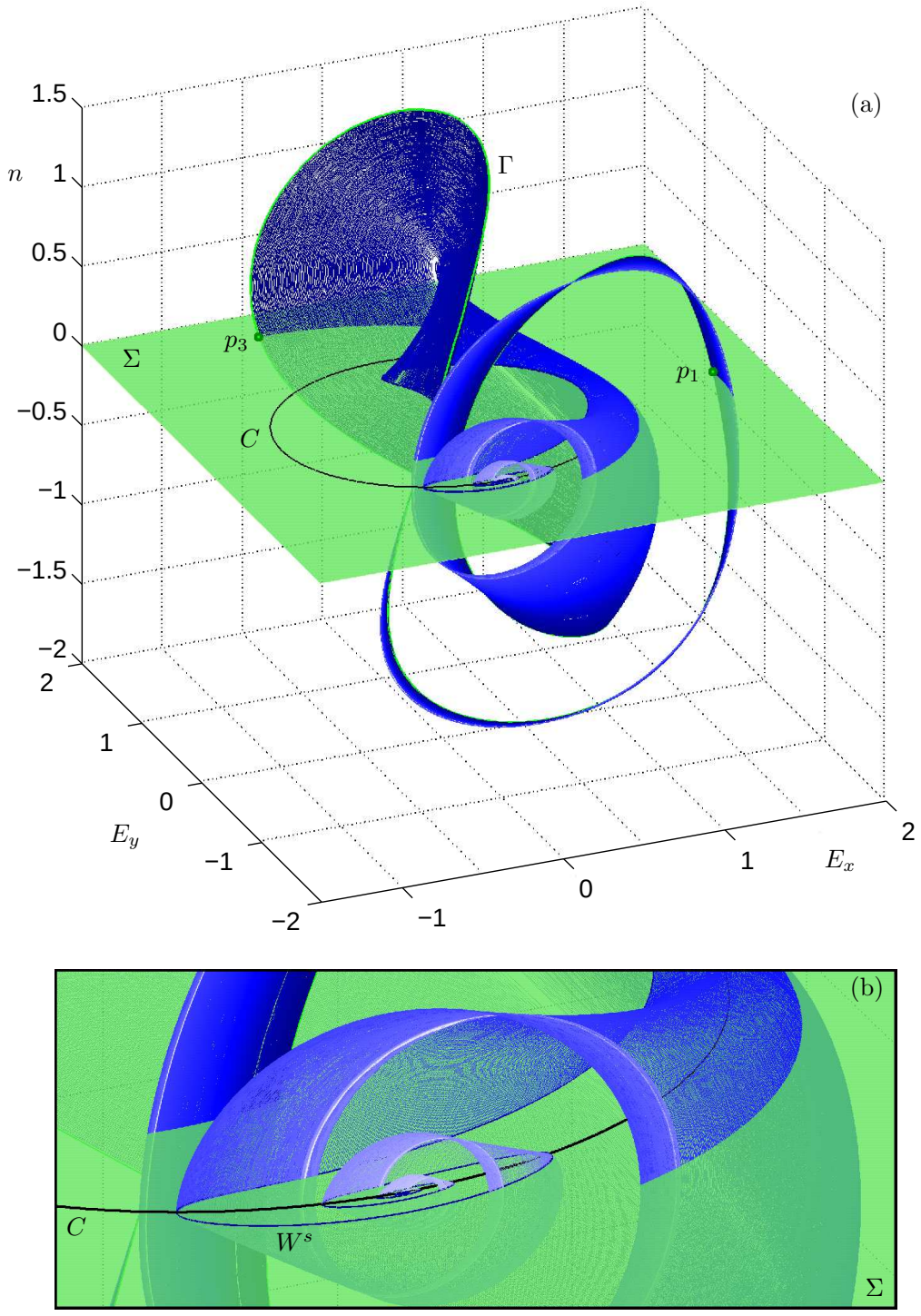


FIG. 5.5. *The orbit segments used in the computation of the right branch of $W^s(p_2)$ are shown in (a) with an enlargement shown in (b). Each time the manifold crosses C, the orbit segment used to compute $W^s(p_2)$ gains another intersection point with Σ. The color of the orbit segments get lighter as the computation progresses.*

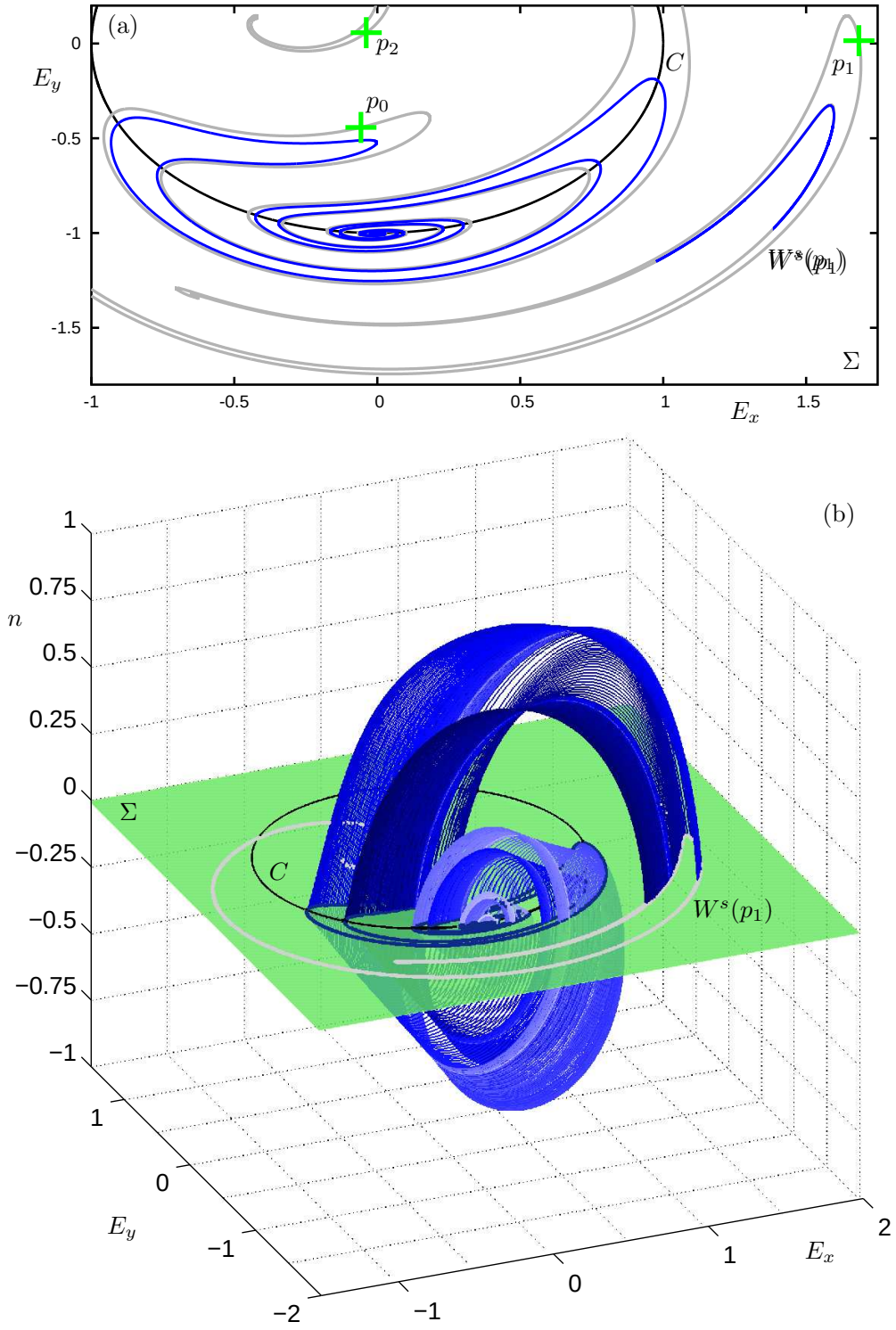


FIG. 5.6. *An enlargement of Figure 5.1(d) in panel (a) shows a disjoint intersection of* $W^s(\Gamma)$ *with* Σ *(blue). The blue section of* $W^s(p_1)$ *maps to this disjoint piece. The orbit segments used in the computation are shown in (b). The boundary point* $\mathbf{u}(0)$ *of the orbit segments lie on* $W^s(p_1)$*, and the end boundary points* $\mathbf{u}(1)$ *form the disjoint piece. See also the animation* eko_a04.gif.

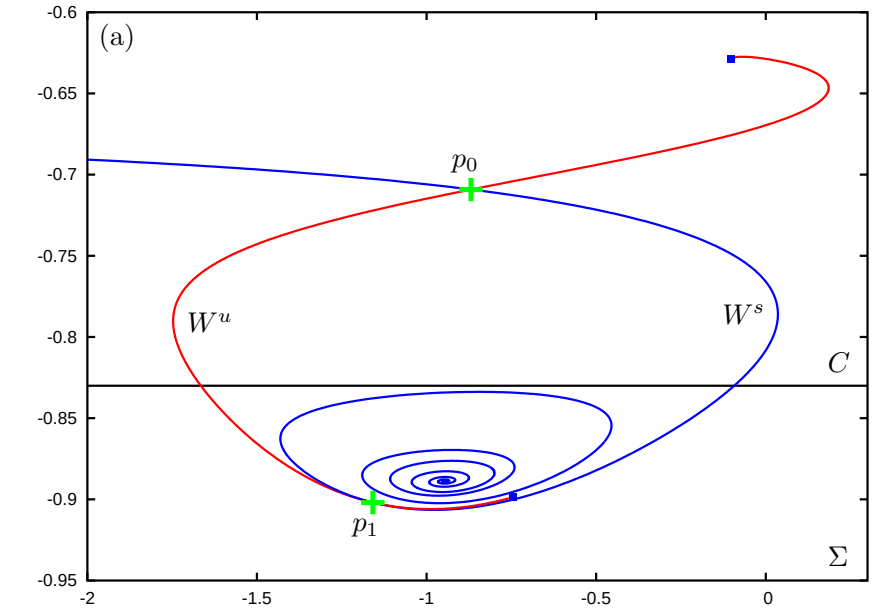
intersects the section Σ. We are able to compute this disjoint piece of W^s by first computing the branch of $W^s(p_1)$. Then, guessing which section of $W^s(\Gamma)$ may have extra intersections with Σ, we choose a computed point on the branch of $W^s(p_1)$ as our starting point. This point is then integrated backward in time until it intersects Σ again, which is a point on the disjoint manifold. We then correct the solution such that it satisfies the boundary conditions and we can use the ManBVP algorithm in the usual way. The boundary point $\mathbf{u}(0)$ is continued along a piece of manifold we have already computed, and $\mathbf{u}(1)$ traces out the new branch of manifold.

6. Slow-fast model of a chemical oscillator. In this section we compute stable and unstable manifolds of a Poincaré map of a slow-fast system. The slow-fast dynamics is characterized by two different time scales, which results in great sensitivity with respect to the initial conditions; see, for example, [13]. The Floquet multipliers of the saddle periodic orbit differ by several orders of magnitude. It is, therefore, already difficult to find the linear eigendirection of the corresponding saddle of the Poincaré map. Indeed, the boundary point $\mathbf{u}(0)$ remains virtually unchanged close to p_0 in the section Σ, while its image $\mathbf{u}(1)$ traces out the entire manifold. Specifically, we consider a Van der Pol-Duffing-type model [1, 14], that has been used in the development of chemical oscillators [7]. The model is given by the equations

$$\begin{aligned} \dot{x} &= \varepsilon_1^{-1}\left(ky - x^3 + 3x - \lambda\right), \\ \dot{y} &= x - 2y + z, \\ \dot{z} &= \varepsilon_2\left(y - z\right), \end{aligned} \tag{6.1}$$

where k and λ are general parameters, and ε_1 and ε_2 are parameters relating to the different timescales of the reaction. We choose the timescale parameters $\varepsilon_1 = 0.1$ and $\varepsilon_2 = 1$, such that the x-variable evolves on a timescale ten times faster than y and z. A saddle periodic orbit Γ and an attracting period-doubled orbit exist for $k = -22.5$ and $\lambda = 18$. These orbit segments were found by continuation from stable solutions using AUTO [4]. We choose the Poincaré section given by the plane $\Sigma = \{(x, y, z) : z = -0.83\}$; the periodic orbit Γ intersects Σ at $p_0 \approx (-0.869, -0.709)$ and $p_1 \approx (-1.158, -0.902)$. For the given parameters, the Poincaré map is discontinuous along the line C given by $y = -0.83$. The eigenvalues of p_0 and p_2 are $\lambda^s = -9.25 \times 10^{-4}$ and $\lambda^u = -3.02843$, computed by solving the first variational equations with AUTO. The eigenvectors of p_0 and p_1 computed by projecting the Floquet vectors down into the section; see section 3.2. Due to the difference in magnitude of the eigenvalues, λ^u is of order 1 and λ^s is of order 10^{-3}, there is considerable contraction along the stable manifold. Notice that the eigenvalues are negative. In order to preserve orientation when growing the manifolds, we must consider the second full return, so that the orbit returns to the correct side of the manifold. In the present setup, this means that the continuation is started from the double covering of the periodic orbit Γ. The contraction along the stable manifold is then $1/(\lambda^s)^2 \approx 1.17 \times 10^6$. It is precisely this difference in scales in slow-fast nature that makes it impossible to compute the Poincaré map by solving an initial value problem; very small changes in the starting conditions will lead to very different orbit segments that may not even return to Σ. By monitoring variations along the entire orbit segment using a boundary value problem formulation, we can effectively compute $\mathbf{u}(1)$ and, hence, compute the manifolds by continuation.

We compute the stable and unstable manifolds of p_0 and p_1 in Σ; see Figure 6.1(a). The saddles p_0 and p_1 are indicated by green crosses. The period-doubled attractor intersects Σ only twice, at the points indicated by blue squares, which are



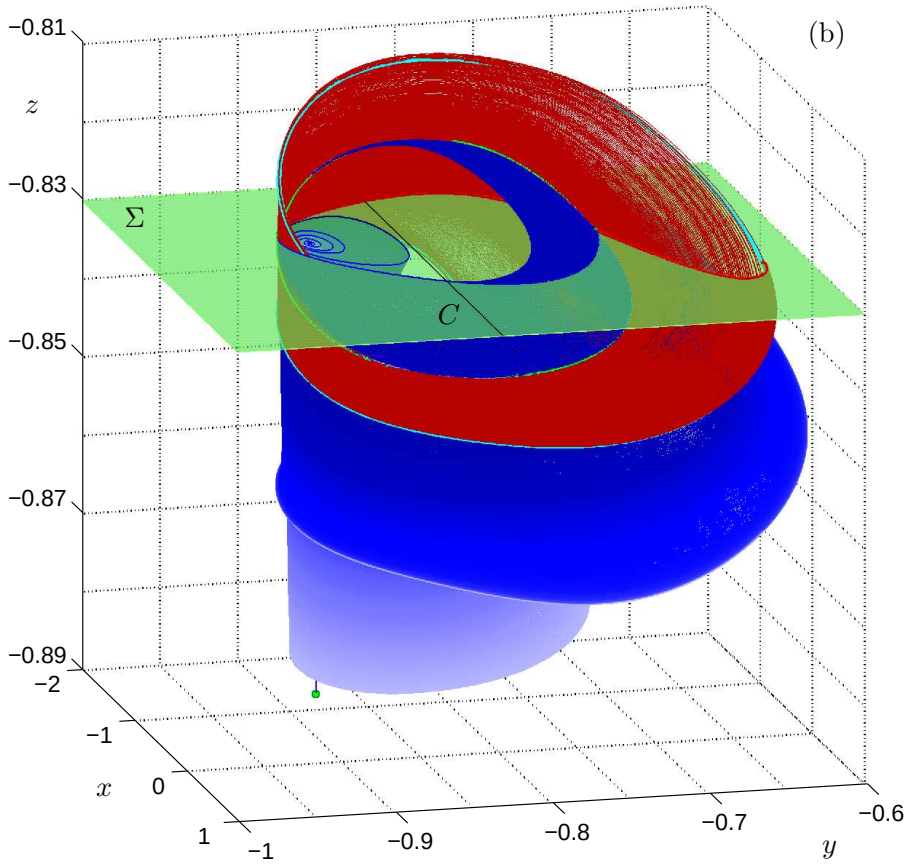


FIG. 6.1. *The stable (blue) and unstable (red) manifolds of (6.1) in the section $z = -0.83$ for parameters $\varepsilon_1 = 0.1$, $\varepsilon_2 = 1$ and $\lambda = 18$, computed using the ManBVP algorithm. The points p_0 and p_1 are the two intersections of a saddle periodic orbit Γ with Σ. The orbit segments used in the computation are shown in panel (b).*

period-two sinks of the associated Poincaré map. The left branch of $W^u(p_0)$ and the right branch of $W^s(p_0)$ both cross C and connect to p_1 (in the same way as for the laser example in Section 5). The right branch of $W^u(p_0)$ tends to one of the sinks and the right branch of $W^u(p_1)$ tends to the other sink. The left branch of $W^s(p_0)$ tends to $-\infty$, whilst the left branch of $W^s(p_1)$ spirals in toward a single point at approximately $(0.946, -0.889, -0.830)$. At this point the Poincaré map is undefined as it is the intersection of the stable manifold of a saddle equilibrium located approximately at $(-0.887, -0.887, -0.887)$, indicated by a green point. The saddle has two complex conjugate unstable eigenvalues, which causes $W^s(\Gamma)$ to spiral around the stable manifold. Figure 6.1(b) shows all orbit segments that were used in the computation of the manifolds.

We demonstrate in Figure 6.2 how $W^s(p_0)$ is grown through C. The situation is very similar to that in Figure 5.3, but the stretching of the stable manifold is extremely large due to the slow-fast nature of the system. In panel (a) the periodic orbit Γ is shown with its two intersections p_0 and p_1 in Σ. Panel (b) shows the manifold grown up to the discontinuity boundary C. All orbit segments start near p_0 and flow backward in the direction $z < -0.83$ from the section. Every orbit shown makes two 'turns'; the first turn lies almost on the saddle periodic orbit, and intersects Σ on the opposite side of, but very close to p_0. Only two-thirds of the way round the second turn, do the orbit segments 'fan out' as $\mathbf{u}(1)$ traces out $W^s(p_0)$. One can see how the last orbit segment computed is tangent to Σ. Panel (c) shows $W^s(p_0)$ computed past C. The boundary points of the orbit segments have now moved back along $W^s(p_0)$ toward p_0 and the orbit segments intersect Σ one extra time. Panel (d) shows the manifold computed up to its connection with p_1. Note that if we had computed $W^s(p_1)$ from p_1 to p_0 instead, we would have got the same piece of manifold using similar orbit segments. However, as $W^s(p_1)$ crosses C, the number of intersections of computed orbit segments with Σ decreases by one.

The structure of the different segments of manifold are shown in detail in Figure 6.3. The left column shows all orbit segments used in the computation, whilst a selection of orbit segments are shown in the right column for a 'see through'. Row (a) shows the orbit segments of the piece of manifold shown in Figure 6.2. The boundary points $\mathbf{u}(1)$ that form the segment of $W^s(p_0)$ past C, are on orbit segments that first intersect Σ along $W^s(p_0)$ before C, and are integrated for longer time until they intersect Σ again. Row (b) of Figure 6.3 shows how the orbit segments of $W^s(\Gamma)$ spiral around the stable manifold of a saddle equilibrium. This is similar to the situation in Section 5, resulting in $W^s(p_1)$ spiraling into a single point in Σ, but in this case the stable manifold of the equilibrium below Σ intersects Σ almost orthogonally. Panel (b2) clearly shows the complicated spiraling of the orbit segments. The orbit segments used in the computation of $W^u(p_0)$ and $W^u(p_1)$ are shown in row (c) of Figure 6.3. Observe how the unstable orbit segments that are part of $W^u(\Gamma)$ tend to a period-doubled attractor (cyan) that intersects Σ only twice. Notice how the 'second loop' of this orbit just misses Σ. The orbit segments give a very good idea of the geometry of the two-dimensional unstable manifold, which due to the negative Floquet multipliers of Γ, forms a Möbius strip.

The orbit segments needed for the computation of the branch of $W^s(p_1)$ that spirals in toward the single point get longer and more complicated as the manifold is grown. Figure 6.4 shows the last orbit (black) of the computation of the left branch of $W^s(p_1)$. The boundary point $\mathbf{u}(0)$ still lies very close to p_0 (2.198×10^{-7}), and the orbit segment closely follows Γ for the first one and a quarter turn. Only then does it

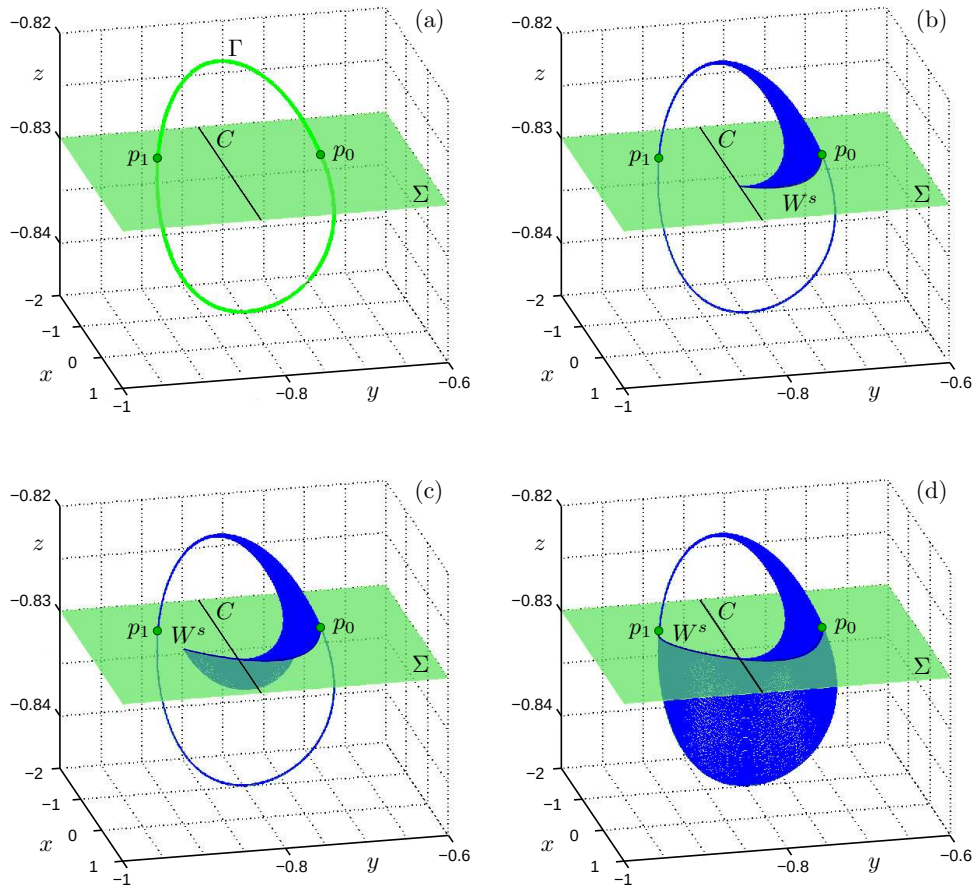


FIG. 6.2. *A demonstration of how one branch of $W^s(p_0)$ for system (6.1) is grown in Σ across C. Panel (a) shows the periodic orbit Γ (green) with two intersections p_0 and p_1 with Σ. Panel (b) shows $W^s(p_0)$ grown up to the point where it touches C. Panel (c) shows the manifold computed past C; $\mathbf{u}(0)$ now moves backward along the manifold and further orbit segments have an extra intersection with Σ. Panel (d) shows the manifold grown up to p_1. All orbit segments make at least two full turns, the first being very close to Γ. They only 'fan out' after about one and two-third turns. See also the animation* eko_a05.gif.

move away from Γ and spirals around the stable manifold of the saddle equilibrium. This highlights the extreme sensitivity with respect to the initial condition. Notice that the last orbit segment is now much longer than the original periodic orbit. For this reason, the number of mesh points is automatically increased when AUTO has difficulty to converge during the computation. While Γ is discretized with 100 mesh points, for the last orbit segment the number of mesh points has increased to 250. The time profiles for each of the three state variables of the last orbit segment computed are shown in panels (b)–(d). The mesh points are indicated by black points and the green line in panel (d) is the Poincaré section Σ. Compared with the time profiles in Figure 5.3, one notices how many more mesh points are needed.

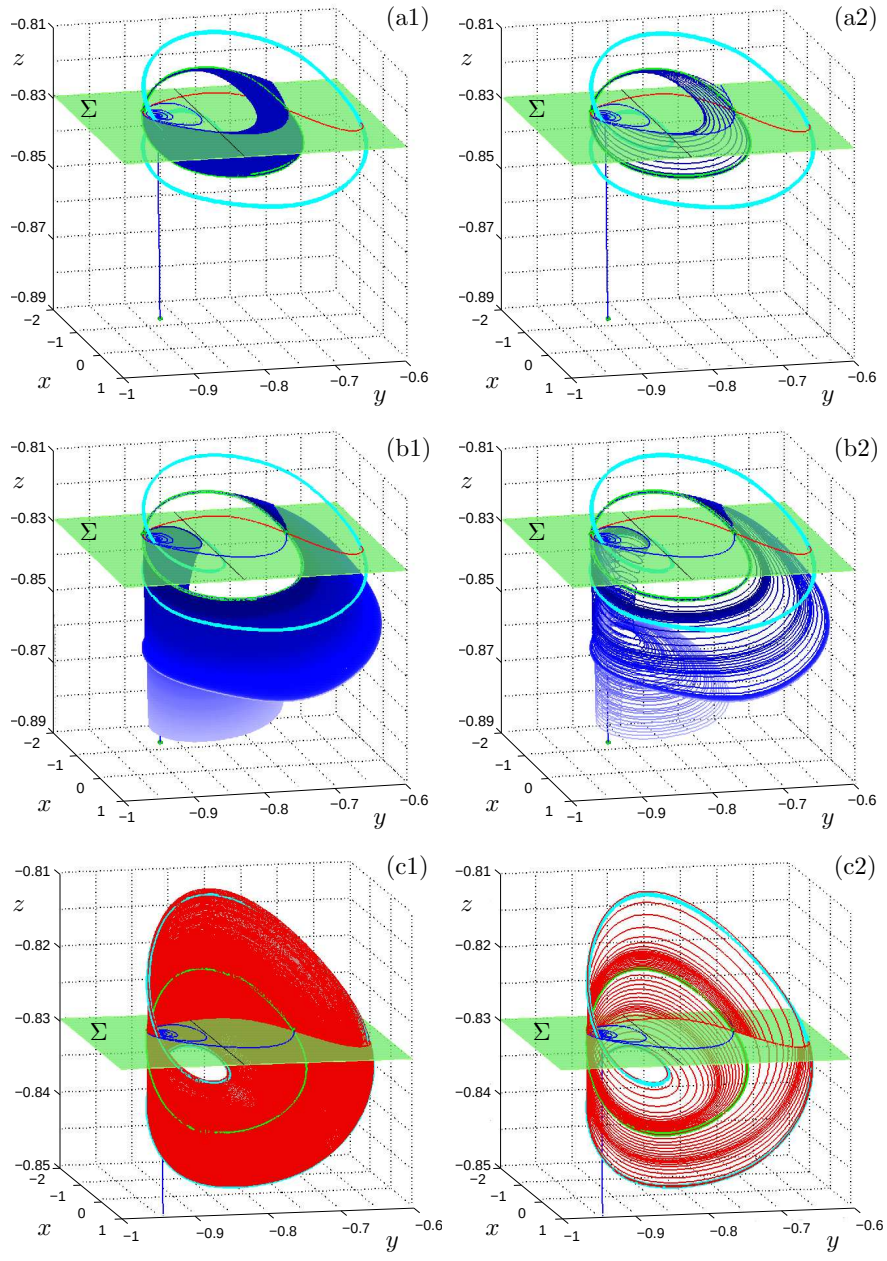


FIG. 6.3. *Three different parts of manifolds are shown for system (6.1). The orbit segments of all mesh points used in the computation are shown in the left column, whilst only a selection are shown in the right column. Row (a) shows the branch of $W^s(p_0)$ shown in Figure 6.2 that connects p_0 and p_1. Row (b) shows the branch of $W^s(p_1)$ that spirals in around the stable manifold of a saddle equilibrium that does not lie in Σ. Row (c) shows both branches of $W^u(p_0)$ and $W^u(p_1)$ and the corresponding orbit segments. Note how the manifolds converge to an attracting period-doubled orbit (cyan); the red surface is a Möbius strip. See also the animation* eko_a06.gif *to see how the part of the manifold shown in row (b) is grown.*

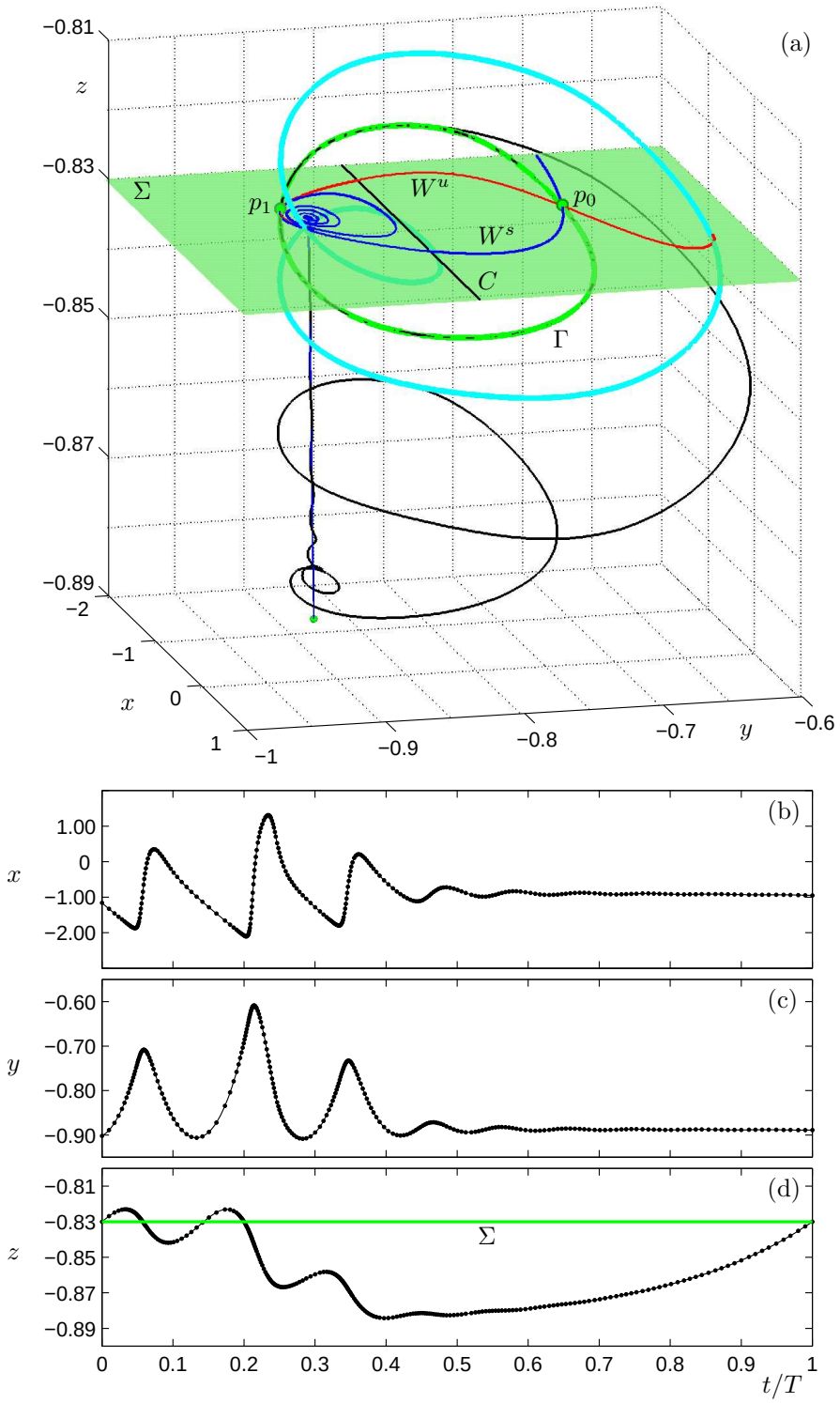


FIG. 6.4. *The last orbit segment (black) used in the computation of the left branch of $W^s(p_1)$ spirals around the stable manifold of the saddle equilibrium (blue line); compare also Figures 6.3(c) and (d). The time profiles for the three state variables of this orbit segment are shown in panels (b)–(d). The green line in panel (d) is the Poincaré section Σ at $z = -0.83$. The black dots are the 250 mesh points used in the* AUTO *collocation.*

7. Conclusions. The ManBVP algorithm presented here is a general-purpose method to compute one-dimensional manifolds of a Poincaré map P of a vector field. In particular, it can compute such manifolds even in situations where standard methods break down — across discontinuity boundaries of the Poincaré map, and when $P(x)$ depends extremely sensitively on the initial condition x, a situation that is typical in slow-fast systems. The performance of the ManBVP algorithm was demonstrated and illustrated with three examples of increasing complexity.

While the present implementation is for three-dimensional vector fields and, hence, Poincaré maps defined on a two-dimensional section, the ManBVP algorithm is not limited to this case. It can be used to compute one-dimensional manifolds of a Poincaré map defined by any codimension-one section. Furthermore, the section need not be a hyperplane, but can be any codimension-one submanifold of the phase space of the vector field on which a Poincaré map can usefully be defined.

The setting itself of following the solution of a two-point boundary value problem can be used in more general situations. For example, the method in [19, 20] for computing two-dimensional (un)stable manifolds of vector fields at present solves boundary value problems by shooting. A future version of this method will also use AUTO's continuation and collocation routines. Note that Doedel [21] also uses the continuation of solutions of a boundary value problem. His method computes two-dimensional (un)stable manifolds of vector fields in AUTO by following an orbit of, say, a particular arclength, while one boundary point is allowed to vary along a circle (or an ellipse) around a saddle point or saddle-periodic orbit.

An entirely more challenging problem is that of computing a two-dimensional (un)stable manifold of a general Poincaré map (not of the vector field). It is a natural approach to generalize the method for diffeomorphisms in [16] in a similar way. While this sounds conceptually quite straightforward, this would require varying the boundary conditions over simplices, not line segments. The implimentation of such a method remains an interesting challenge for future work.

Finally, the bifurcation theory of non-globally defined Poincaré maps is an interesting topic of our ongoing research. The main question is to identify and classify bifurcations that are due to interactions with the discontinuity boundary. The ManBVP algorithm makes it possible to study such bifurcation that involve (un)stable manifolds. Note that this research is closely related in spirit to the study of global bifurcations in noninvertible maps [6], for which we developed the SearchCircle method [5] that computes one-dimensional stable manifolds in the absence of an inverse.

8. Acknowledgments. We are grateful to Sebius Doedel for his help with getting the ManBVP algorithm to communicate with AUTO. Furthermore, we thank Pieter Collins for helpful discussions on non-globally defined Poincaré maps. J.E. was supported by grant GR/R94572/01 from the Engineering and Physical Sciences Research Council (EPSRC), and B.K. by an EPSRC Advanced Research Fellowship grant.

REFERENCES

[1] J. Boissonade and P. DeKepper, *Transition from bistability to limit cycle oscillations*, J. Phys. chem. **84** (1980) pp. 501–506.

[2] H. W. Broer, B. Krauskopf and G. Vegter (Eds.), *Global analysis of dynamical systems*, Festschrift dedicated to Floris Takens for his 60th birthday, IoP Publishing, Bristol, 2001.

[3] P. Collins and B. Krauskopf, *Entropy and bifurcation in a chaotic laser*, Phys. Rev. E **66**, 056201 (2002).

[4] E. Doedel, *AUTO, software for continuation and bifurcation problems in ordinary differential equations*, Department of Applied Mathematics, California Institute of Technology, (1986).
[5] J. P. England, B. Krauskopf and H. M. Osinga, *Computing one-dimensional stable manifolds and stable sets of planar maps without the inverse*, SIAM J. Appl. Dyn. Sys. 3(2) (2004) pp. 161–190.
[6] J. P. England, B. Krauskopf and H. M. Osinga, *Bifurcations of stable sets in noninvertible planar maps*, Int. J. Bifurcation and Chaos **15**(3) (2005), to appear.
[7] I. R. Epstein, K.Kustin, P. DeKepper and M. Orban, *Oscillating chemical reactions*, Sci. Am. **3** 248 (1983) pp. 112–123.
[8] K. Green, B. Krauskopf and K. Engelborghs, *One-dimensional unstable eigenfunction and manifold computations in delay differential equations*', Journal of Computational Physics, Vol 197 (2004) pp. 86–98.
[9] J. Guckenheimer and P. Holmes, *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, Springer-Verlag, NY, second printing, 1983.
[10] J. Guckenheimer, K. Hoffman and W. Weckesser, *Global bifurcations of periodic orbits in the forced Van der Pol equation*, in [2].
[11] M. W. Hirsch and S. Smale, *Differential equations, dynamical systems, and linear algebra*, Academic Press, Inc., 1974.
[12] D. Hobson, *An efficient method for computing invariant manifolds*, Journal of Computational Physics, Vol 104 (1991) pp. 14–22.
[13] C. K. R. T. Jones, *Geometric singular pertubation theory*, C.I.M.E. Lectures, Montecatini Terme, June 1994, Lecture Notes in Mathematics **1609**, Springer-Verlag, Heidelberg, 1995.
[14] M. T. M. Koper, *Bifurcations of mixed-mode oscillations in a three-variable autonomous Van der Pol-duffing model with a cross-shaped phase diagram*, Physica D 80 (1995) pp. 72–94.
[15] A. Kovanis, A. Gavrielides, T. Simpson and J. Liu, *Instabilities and chaos in optically injected semiconductor lasers*, Appl. Phys. Lett. **67** (19) (1995) pp. 2780–2782.
[16] B. Krauskopf and H. M. Osinga, *Globalizing two-dimensional unstable manifolds of maps*, Int. J. Bifurcation and Chaos **8** (3) (1998) pp. 483–503.
[17] B. Krauskopf and H. M. Osinga, *Growing 1D and quasi-2D unstable manifolds of maps*, Journal of Computational Physics, Vol 146 (1998) pp. 406–419.
[18] B. Krauskopf and H. M. Osinga, *Investigating torus bifurcations in the forced Van der Pol oscillator*, in Numerical Methods for Bifurcation Problems and Large-Scale Dynamical systems, E. J. Doedel and L. S. Tuckerman, eds., IMA Volumes in Mathematics and its Applications, Springer-Verlag, New York, 1999.
[19] B. Krauskopf and H. M. Osinga, *Two-dimensional global manifolds of vector fields*, CHAOS **9**(3) (1999) pp. 768–774.
[20] B. Krauskopf and H. M. Osinga, *Computing geodesic level sets on global (un)stable manifolds of vector fields*, SIAM J. Appl. Dyn. Sys. 2(4) (2003) pp. 546–569.
[21] B. Krauskopf, H. M. Osinga, E. J. Doedel, M. E. Henderson, J. Guckenheimer, A. Vladimirsky, M. Dellnitz and O. Junge, *A survey of methods for computing (un)stable manifolds of vector fields*, Int. J. Bifurcation and Chaos **15**(3) (2005), to appear.
[22] Yu. A. Kuznetsov, *CONTENT - integrated environment for analysis of dynamical systems. Tutorial.*, Ecole Normale Superieure de Lyon, Rapport de Recherche UPMA-98-224 (1998).
[23] J. Palis and W. de Melo, *Geometric Theory of Dynamical Systems*, Springer-Verlag, New York/Berlin, 1982.
[24] T. S. Parker and L. O. Chua, *Practical Numerical Algorithms for Chaotic Systems*, Springer-Verlag, Berlin, 1989.
[25] B. Van der Pol Radio Rev. 1, (1920) pp. 704–754.
[26] A. Sherman, *Calcium and membrane potential oscillations in pancreatic beta-cells*, in H. G. Othmer, F. R. Adler, M. A. Lewis and J. C. Dallon (Eds.), *Case Studies in Mathematical Modeling - Ecology, Physiology, and Cell biology*, Prentice-Hall, NJ, 1997 pp. 199–217.
[27] S. H. Strogatz *Nonlinear dynamics and chaos: With applications to physics, biology, chemistry, and engineering* Reading, MA: Perseus Books, Cambridge MA, 1994.
[28] S. M. Wieczorek, B. Krauskopf and D. Lenstra, *A unifying view of bifurcations in a semiconductor laser subject to optical injection*, Opt. Commun. **172** (1999) pp. 279–295.
[29] S. M. Wieczorek, B. Krauskopf, T. B. Simpson and D. Lenstra, *The dynamical complexity of optically injected semiconductor lasers*, Phys. Reports, to appear.
[30] Xiao-Song Yang, *A remark on global Poincaré section and suspension manifold*, Chaos, Solitons and Fractals 11 (2000) pp. 2157–2159.
[31] Z. You, E. J. Kostelich and J. A. Yorke, *Calculating stable and unstable manifolds*, Int. J. Bifurcation and Chaos, Vol 1, No. 3 (1991) pp. 605–623.